

[논문읽기] A Style-Based Generator Architecture for Generative Adversarial Networks

🕒 4 minute read

이 글은 2019 CVPR 논문, A Style-Based Generator Architecture for Generative Adversarial Networks (<https://arxiv.org/abs/1812.04948>)를 참고하여 작성하였습니다.

NVIDIA의 StyleGAN 논문 리뷰로 첫 블로그 글을 작성하게 되었네요.

이미 올해 초에 큰 화제가 되었고, CVPR에서도 큰 관심을 받았던 논문이라고 들었습니다. 그만큼 다른 리뷰글들이 이미 많이 있지만, 저도 그 글들을 참고해서 한번 리뷰해보겠습니다.)



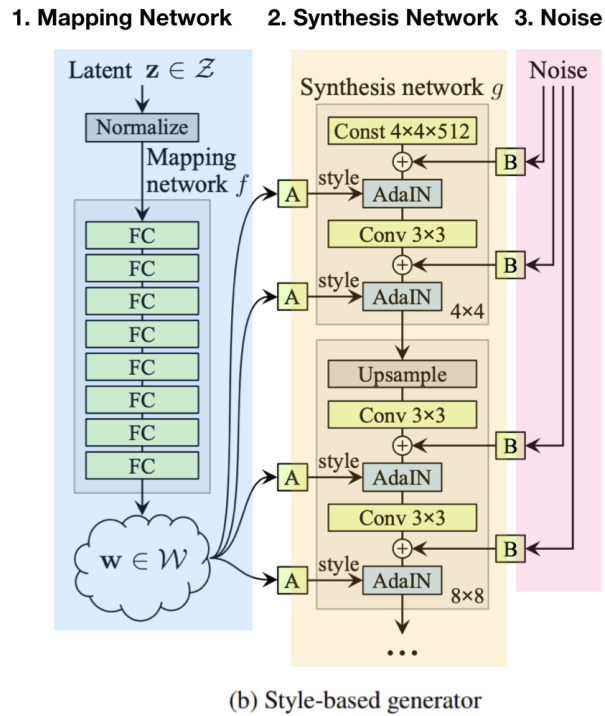
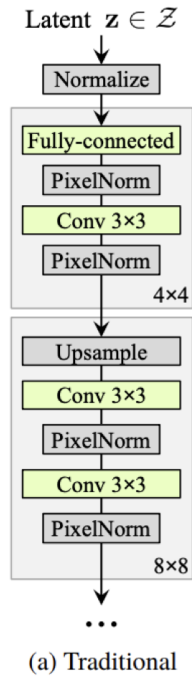
StyleGAN을 통해 생성된 이미지들(가짜 이미지들)을 보면, 굉장히 고해상도/고퀄리티의 이미지를 볼 수 있습니다. 이 논문에서는 기존에 Traditional한 네트워크에서 여러가지를 개선하여서 더 좋은 이미지 생성을 가능하게 했습니다. 또한, 이전에는 가능하지 않았던 "scale-specific control"을 가능하게 했습니다.

이 논문의 특징들은 다음과 같이 크게 3가지로 정리할 수 있습니다.

1. scale-specific control of the synthesis
2. separation of high-level attributes and stochastic variation
3. new metrics: perceptual path length, linear separability

그럼, 어떠한 방법을 이용했는지 한번 살펴보도록 하겠습니다.

Style-based generator



위에 그림은 논문에 실려있는 그림을 가져온 것입니다. 왼쪽이 traditional network (progressive GAN 인 것 같습니다), 오른쪽이 이 논문에서 제안한 Style-gased generator 입니다. 왼쪽 네트워크와 오른쪽에 Synthesis Network가 똑같은 구조를 갖고 있지만, 이전 GAN에서는 latent z 를 바로 input으로 넣어줬던 것과는 다르게, StyleGAN에서는 학습된 constant 값을 넣어줍니다. 또한, 새롭게 Mapping Network와 Noise가 추가된 것을 볼 수 있습니다.

1. Mapping Network가 새롭게 생김... z 에서 w 를 매핑한다?
2. Synthesis Network? AdaIN, style ... 에 대해서
3. Noise는 뭘 뜻하는가?

1. Mapping Network

StyleGAN에서는 Latent z 를 곧바로 Synthesis Network에 넣어주는 것이 아니라, Mapping Network를 통해 w 를 매핑한 후, w 를 넣어줍니다. 논문에서는 이것에 대한 이유를 disentanglement라고 설명합니다.

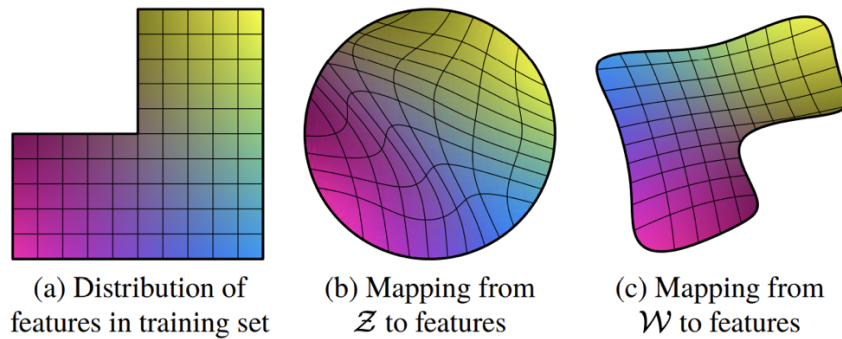


Figure 6. Illustrative example with two factors of variation (image features, e.g., masculinity and hair length). (a) An example training set where some combination (e.g., long haired males) is missing. (b) This forces the mapping from \mathcal{Z} to image features to become curved so that the forbidden combination disappears in \mathcal{Z} to prevent the sampling of invalid combinations. (c) The learned mapping from \mathcal{Z} to \mathcal{W} is able to “undo” much of the warping.

다음은 Disentanglement에 관하여 논문에 실린 그림입니다. 여기서 Disentanglement를 이 논문에서는

“latent space that consists of linear subspaces, each of which controls one factor of variation”

라고 정의합니다.

잘 와닿지 않아서, [Jaeyun's Blog](https://jayhey.github.io/deep%20learning/2019/01/14/style_based_GAN_1/) (https://jayhey.github.io/deep%20learning/2019/01/14/style_based_GAN_1/)를 참고하였습니다. 예를 들어 z 의 특정한 값을 바꿨을 때 생성되는 이미지의 하나의 특성(성별, 머리카락의 길이, 바라보는 방향 등)만 영향을 주게 되면 disentanglement라고 합니다.

원래 기존의 네트워크에서는 z 를 바로 input으로 넣어주는데, 그러면 고정된 z 의 분포를 training set 분포에 맞추려고 하기 때문에 disentangled 하지 못하게 됩니다. 위 그림에서도 볼 수 있듯이 만약 training set에 missing combination이 있다면, Z 는 이러한 이상한 combination이 나오지 않도록 하기 위해 curved 되서 entangled하게 되죠.

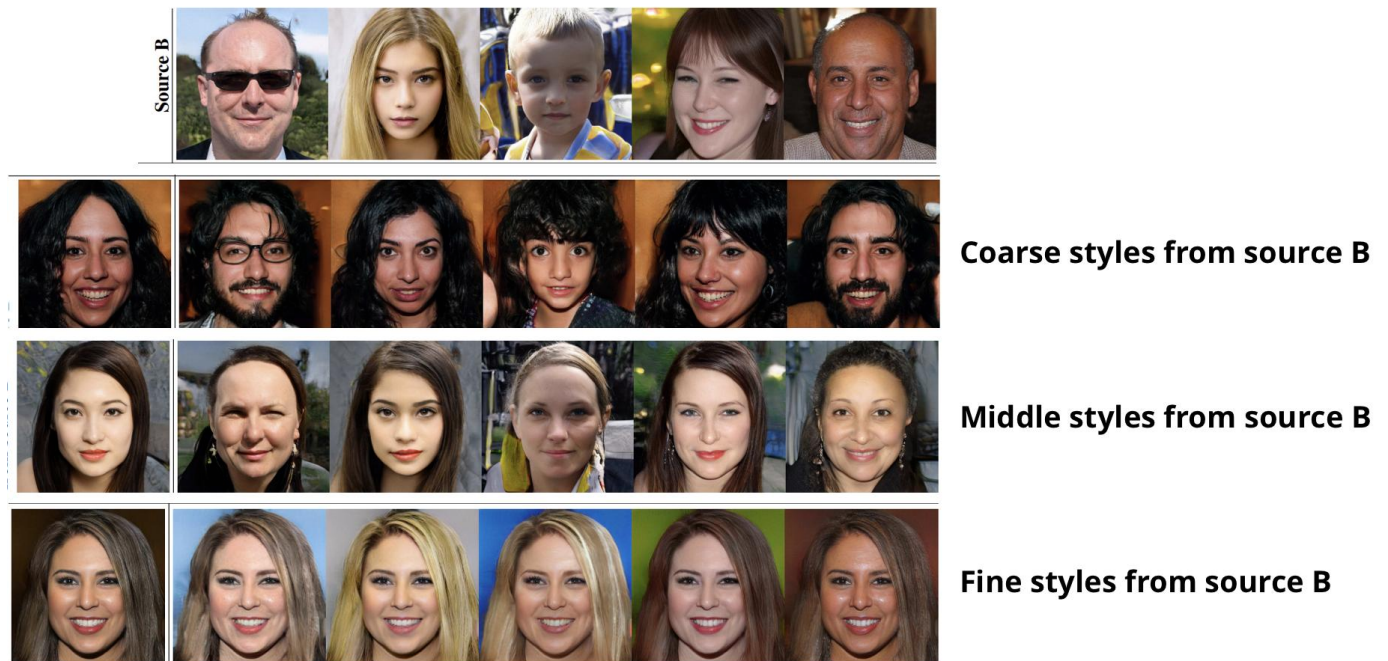
하지만, W 를 feature에 매핑하는 경우에는 다릅니다. W 는 Z 처럼 고정된 분포를 따르지 않습니다. sampling density는 학습된 piecewise continuous mapping $f(z)$ (f 는 mapping network 입니다)에 의해 정해지게 됩니다. 따라서, warping(틀어짐)이 많이 일어나지 않습니다. 그렇기 때문에 factors of variation은 더욱 linear하고, disentangled 하다고 할 수 있습니다. 이것이 바로 z 를 곧바로 feature에 매핑하는 것보다 w 에 매핑하는 것의 장점입니다.

2. Synthesis Network

z를 중간 latent space W에 매핑을 한 뒤에 이 w는 "A"를 거쳐서 style, $y = (y_s, y_b)$ 로 변형됩니다. 이때 A는 학습된 affine transform 입니다. 그리고 이 style들은 AdaIN(adaptive instance normalization) operation을 control 합니다.

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

AdaIN (<https://arxiv.org/pdf/1703.06868.pdf>)은 style transfer를 할 때 많이 쓰이는 방법으로, 임의의 style transfer를 실시간으로 가능하게 합니다. (나중에 이 논문도 읽어봐야겠네요.) 여기서 feature map x_i 는 normalized 된 다음에, style로 변환된 두 y로 scaled, biased 됩니다. style이 입력되는 거죠. 이 과정을 매 layer 마다 반복합니다. 그리고 이러한 방법은 **scale-specific control** 을 가능하게 합니다.



이렇게 각 layer 마다 다른 style 조절이 가능합니다. ($4^2 - 8^2$)에서는 coarse style (pose, hair style, face shape 등) ($16^2 - 32^2$)에서는 middle style (facial features, hair style, eyes open/closed 등) ($64^2 - 1024^2$)에서는 fine style (color scheme) 등을 나타내서 각 부분에 다른 w를 넣어서 style 조절을 가능하게 하죠.

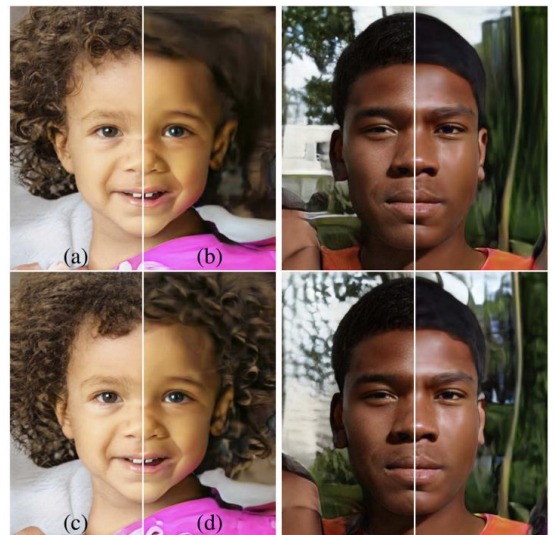
하지만, style 들이 localized 될 수 있도록 하는 거엔 AdaIN 만으로 부족하고, 여기서는 **Style Mixing**이라는 방법도 사용합니다. 간단히 설명하자면, mixing regularization이라고 할 수 있는데, training 때 z를 하나가 아닌 두 개의 z를 넣어주는 것입니다.

3. Noise

위에 Synthesis Network에서 결정하는 것이 high-level 특성들이라고 한다면, 이 noise를 통해 결정하는 것은 stochastic variation 입니다. stochastic variation은 머리의 세세한 결, 콧수염, 주근깨와 같이 perception of the image에 영향을 주지 않고, randomized 될 수 있는 부분들을 말합니다.



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



왼쪽 그림처럼 stochastic variation에 따라서 머리카락의 모양이 미세하게 달라지는 것을 볼 수 있습니다. 또한, 오른쪽 그림처럼 각 layer에 noise를 넣었는지 넣지 않았는지 여부에 따라서도 머리카락 모양의 차이가 생깁니다.

만약 둘을 같은 latent codes z 로 나타내려고 한다면, high-level 특성들은 각 픽셀마다 똑같이 변화해야 하는데, stochastic variation의 경우에는 픽셀별로 다르게 나타나야하기 때문에 잘 표현이 되지 않겠죠. high-level 특성들과 이러한 stochastic variation을 따로 구분하면, per-pixel noise가 가능하게 되고, 더욱 세세한 표현을 할 수 있게 됩니다.

New metrics

이 논문에서는 disentanglement를 측정하는 metric 두 가지를 새롭게 제안합니다. Perceptual path length와 Linear separability입니다.

Method	Path length		Separability
	full	end	
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%	231.5	182.1	3.51
F + Mixing 90%	234.0	195.9	3.79

"more disentanglement!"

1. Perceptual path length

measure how drastic changes the image undergoes as we perform interpolation in the latent space

latent space에서 interpolation을 했을 때, 얼마나 큰 변화가 있는지 측정하는 것입니다. 왜 interpolation을 해서 측정을 하나면, interpolation을 했을 때 일어나는 변화는 disentanglement와 관련이 있기 때문입니다. 예를 들어, interpolation을 했을 때 non-linear한 변화가 이미지에서 일어난다면, latent space가 entangled 할 수 있다는 것이죠.

$$l_Z = \mathbb{E} \left[\frac{1}{\epsilon^2} d(G(\text{slerp}(\mathbf{z}_1, \mathbf{z}_2; t)), G(\text{slerp}(\mathbf{z}_1, \mathbf{z}_2; t + \epsilon))) \right], \quad l_W = \mathbb{E} \left[\frac{1}{\epsilon^2} d(g(\text{lerp}(f(\mathbf{z}_1), f(\mathbf{z}_2); t)), g(\text{lerp}(f(\mathbf{z}_1), f(\mathbf{z}_2); t + \epsilon))) \right],$$

이것을 측정하기 위하여 논문에서는 “perceptually-based pairwise image distance”를 구합니다. 이것의 식은 위와 같습니다. 각각 z와 w 일 경우의 distance를 구하는 공식이고, z는 spherical interpolation을 하고, w는 linear interpolation을 하는데, W의 벡터들은 normalized 되어있지 않기 때문입니다.

2. Linear separability

measure how well the latent-space points can be separated into two distinct sets via a linear hyperplane

latent space가 충분히 disentangled 하다면, 각각의 factors of variation에 해당하는 방향 벡터를 찾을 수 있어야 한다고 합니다. 따라서 latent-space 점들이 linear hyperplane으로 두개로 잘 구분이 되는지를 측정합니다. 이때, 40개의 특성을 갖는 데이터 셋에서 auxiliary classification network를 이용하여 SVM으로 binary classification을 하고, 예측된 class 를 따져서 conditional entropy $H(Y | X)$ 를 구합니다. 이때, X는 SVM으로 예측된 class를, Y는 pre-trained classifier로 예측된 class를 말합니다.


이를 이용해 sample이 true class를 결정하기 위해 additional information이 얼마나 필요한지 알 수 있고, 이로 separability score를 계산하여 metric으로 사용합니다.

이렇게 저의 첫 논문 리뷰를 마쳤습니다. 이 논문을 읽으면서 더욱 GAN에 대해서 공부해보고 싶은 욕구가 생기는 동시에 부족함을 많이 느낍니다. 아직 여기서 나오는 WGAN, 여러 개념들에 대해서 잘 모르는 것들이 많은데, 더 공부해서 또 글을 써봐야겠습니다.

Reference:

- [1] [StyleGAN 논문](https://arxiv.org/abs/1812.04948) (https://arxiv.org/abs/1812.04948).
- [2] [StyleGAN 코드](https://github.com/NVLabs/stylegan) (https://github.com/NVLabs/stylegan).
- [3] [Jaeyun's Blog](https://jayhey.github.io/deep%20learning/2019/01/14/style_based_GAN_1/) (https://jayhey.github.io/deep%20learning/2019/01/14/style_based_GAN_1/).

 **Categories:** Paper-Review

 **Updated:** August 01, 2019

LEAVE A COMMENT