

StyleGAN

[GAN 때문이야] 6주차 발제자료

StyleGAN

18기 분석 김가영

18기 분석 박규연



StyleGAN

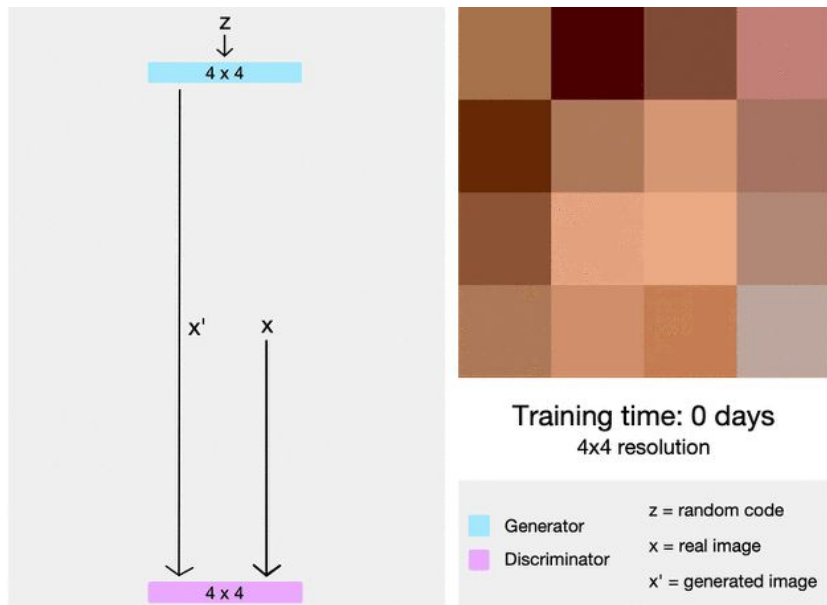


[FFHQ dataset]

PGGAN 구조에서 **Style transfer** 개념을 적용한 새로운 **generator architecture**

- 성별, 포즈같이 큰 특징부터 머리색, 피부톤 등 세밀한 특징까지 조절 가능
- 안정성, 퀄리티 뛰어남

PGGAN



학습을 진행하는 과정에서
점진적으로 **layer**를 추가해나가면서 해상도를
증가시키는 방식의 **generator architecture**

- 4x4 -> 1024x1024 로 feature map을 점차 키워나감
- 한 번에 고해상도 이미지를 만드는 것 보다 훨씬
안정적으로 이미지 생성 가능
- 학습의 속도 또한 굉장히 향상됨

But...

- 이미지의 특징 제어가 어려움

이러한 PGGAN의 한계를 극복하고자 **StyleGAN**에서는 image의 특징들을
disentangle시키는 구조로 특징 제어 성능을 향상시킴

Style Transfer

이미지의 주된 형태는 **content**와 유사하게 유지하면서 **style**만 우리가 원하는 **style image**와 유사하게 바꾸는 것

이미지 == style들의 조합

특정 이미지에서 **style**(나무)을 뽑아내고 다른 이미지에서 **content**(의자)를 뽑아내서 이를 합성 (나무의자)하는 방식

Content target



Style reference



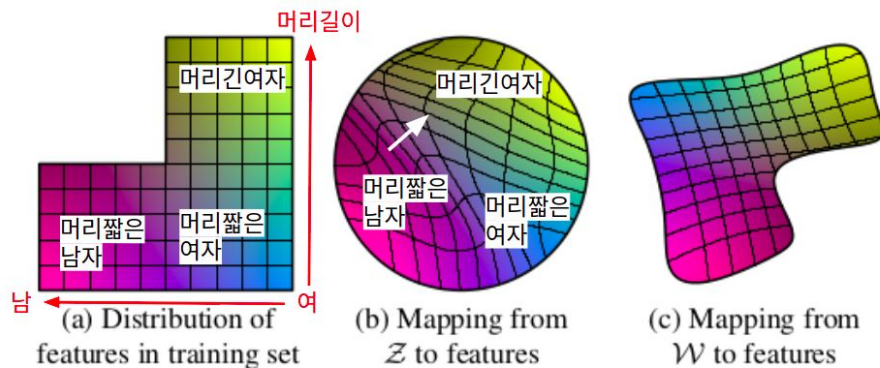
+

=

Combination image



Disentangle



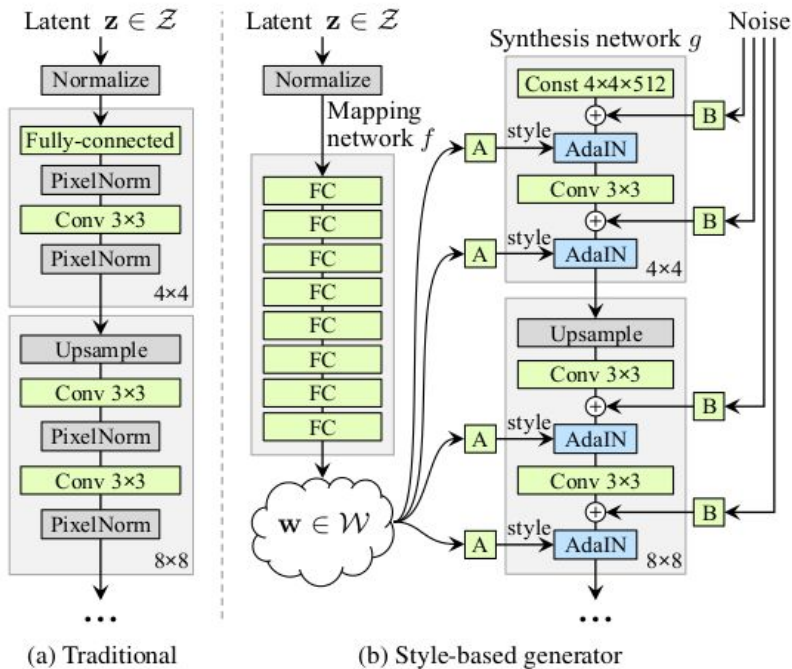
이미지에 포함되어있는 각 특징들이 잘 분리되어있음을 **disentangle**로 나타냄

반면에 나이, 성별, 안경 유무 등의 정보들이 얹혀있다면 **entangle**한 상태

특징들이 **entangle**한 상태라면 특징들을 분리해서 컨트롤하기 어려움

예를 들어, 안경 유무를 조작하려고했는데 성별이 바뀌는 등 원하는 특징의 조작이 어려울 수 있음

Traditional vs StyleGAN architecture



Traditional

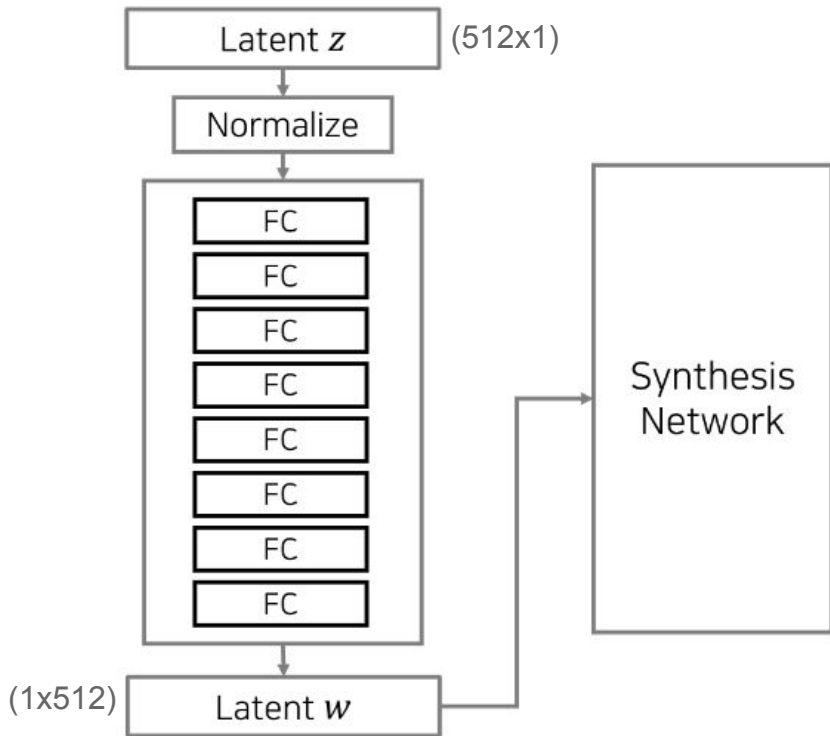
- latent vector인 z 를 직접 input하여 start
- z 가 직접 (4x4)에서 (1024x1024)까지 upsampling 및 convolution을 거쳐 이미지로 변환

StyleGAN

- 학습된 상수 tensor로 Synthesis Network인 g 를 start
- latent vector인 z 를 Mapping Network인 f 에 통과시켜 intermediate latent vector w 로 변환 후 합성 신경망의 각 블록에 각각 input으로 넣어줌

input과 start 방식의 차이 집중~!!!!

Mapping network (8-layer MLP)



샘플링 시, 데이터(Latent space)는
가우시안 분포를 따른다고 가정

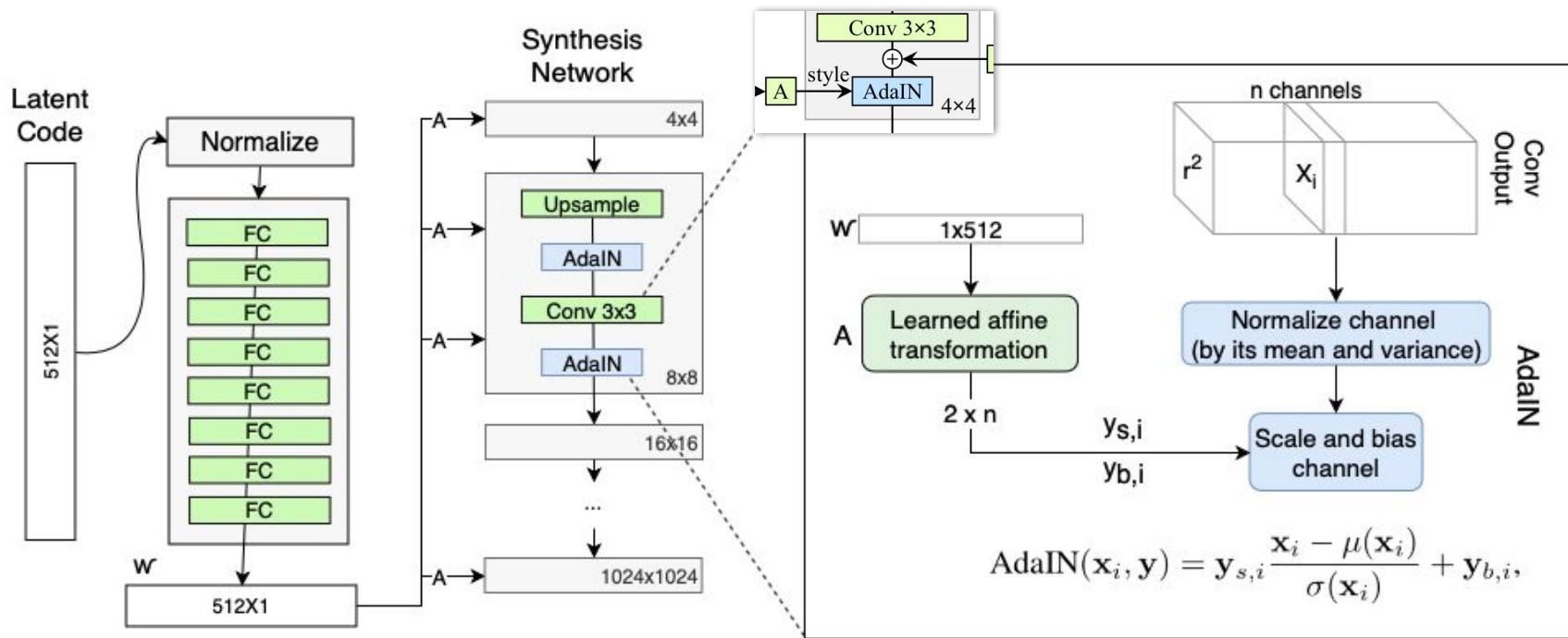
PGGAN

latent vector z 를 직접 input으로 넣어줌
feature들의 **disentanglement**가 보장되지 않음
→ 특징 제어 어려움

StyleGAN

$z \rightarrow w$ 로 mapping함으로써 feature들의
disentanglement 보장

Style Module(AdaIN)



Affine transformation: 선형변환을 한 후 시프팅(Translation)

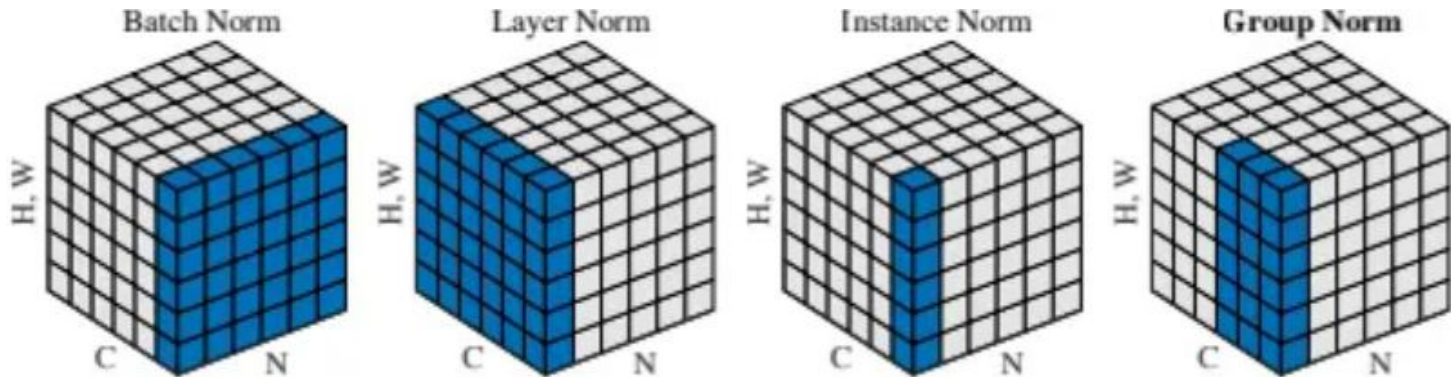
AdaIN(Adaptive Instance Normalization)

Style transfer architecture에서 사용된 normalization 방식

Mapping network를 거쳐나온 latent vector W 가 layer에 입력으로 들어갈 때 마다 style이 입혀질 수 있도록 함

AdaIN에서 정규화를 할 때 한 번에 하나의 W 만 기여하므로 하나의 style이 각각의 scale에서만 영향을 끼칠 수 있도록 분리해주는 효과를 가짐

각각의 이미지, 각 채널에서 하나의 feature 맵에 대해 정규화 수행



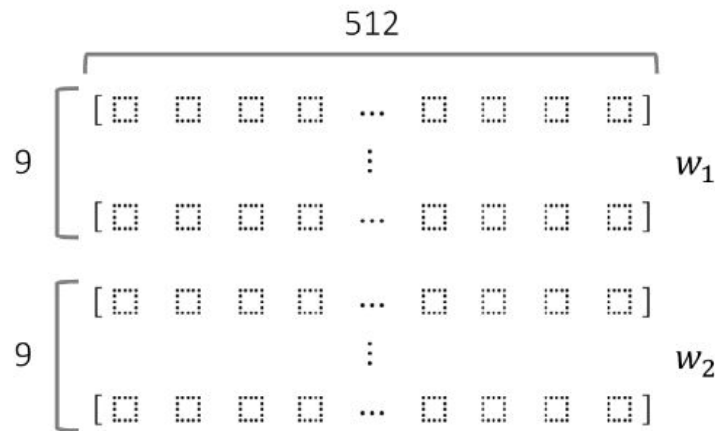
Style mixing

인접한 layer간의 상관관계를 줄이는テクニック

다양한 특징들이 더 잘 분리될 수 있도록 함

- 두 개의 **latent input vector**(w_1, w_2) 준비
- 크로스오버(cross over) 포인트 설정
- 크로스오버 포인트 이전은 w_1 벡터, 이후는 w_2 벡터 사용

이를 통해 **style**은 각 layer에 대하여 **localization**됨



https://github.com/ndb796/Deep-Learning-Paper-Review-and-Practice/blob/master/lecture_notes/StyleGAN.pdf





왼쪽 이미지는 **source B**로 부터 style을 가져와

각각 **source A**의 Coarse, Middle, Fine에 합성한 예시



Coarse style layer(4)

처음 4개의 layer에 style을 적용한 경우, 이후 많은

layer들을 거치기 때문에 이미지의 큰 변화에 영향을 미침



Middle style layer(4)

중간 4개의 layer에 style을 적용한 경우, Coarse 보단 더,

Fine 보단 덜 세밀한 부분의 변화

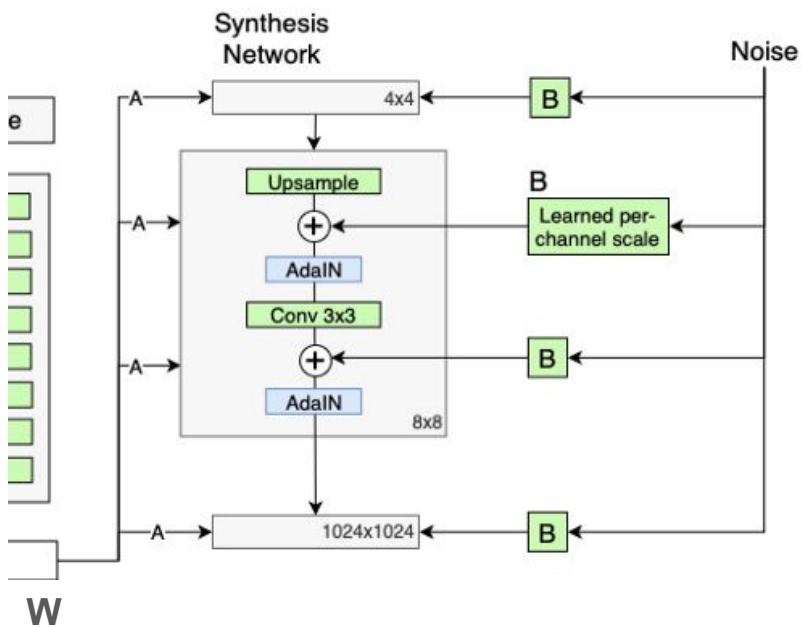


Fine style layer(10)

마지막 10개의 layer에 style을 적용한 경우, 미세한 detail들에 변화

Stochastic variation

별도의 **noise**를 입력으로 넣어주어 하나의 이미지에 포함될 수 있는 다양한 확률적 측면, **stochastic variation**을 컨트롤 함



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

Style(W) / Noise

(a) 모든 레이어에 노이즈 적용

(b) 노이즈 적용X

(c) Fine layer에 노이즈 적용

(d) coarse layer에 노이즈 적용



Style(w)

high-level **global attributes**

- 얼굴형, 포즈, 안경 유무 등

Noise

stochastic variation

- 주근깨, 피부 모공 등

- **Coarse noise:** 큰 크기의 곱슬거림, 배경 등

- **Fine noise:** 세밀한 곱슬거림, 배경 등

Conclusion

기존의 GAN 신경망에 비해서 모든 면에서 성능이 우수

특징들을 linear하게 분리하여 disentangle을 보장하도록 함

고화질 이미지 생성에 용이



References

[원문] <https://arxiv.org/abs/1812.04948>

https://blog.promedius.ai/stylegan_1/

<https://velog.io/@ghgh5317/StyleGAN-v1-%EA%B0%84%EB%8B%A8%ED%95%98%EA%B2%8C-%EC%A0%95%EB%A6%AC>

<https://www.youtube.com/watch?v=HXgfw3Z5zRo>

<https://www.notion.so/AdaIN-Arbitrary-Style-Transfer-in-Real-Time-With-Adaptive-Instance-Normalization-9fe8b5fb60154380b6fbe3147e0afe9e>

<https://blog.lunit.io/2017/04/27/style-transfer/>

https://github.com/ndb796/Deep-Learning-Paper-Review-and-Practice/blob/master/lecture_notes/StyleGAN.pdf

