Progressive Growing of GANs for Improved Quality, Stability, and Variation

한양대학교 AILAB 석사과정 엄희송

ABSTRACT

ICLR 2018
Tero Karras, Timo Aila, Samuli Laine (NVIDIA), Jaakko Lehtinen(NVIDIA & Aalto University)

GAN을 학습시키는 새로운 방법론을 제안해 GAN의 학습 속도, 생성되는 이미지의 안정성과 다양성을 높임.

1. INTRODUCTION

현재 가장 많이 쓰이는 Generative method 에는 3가지 방법이 있음

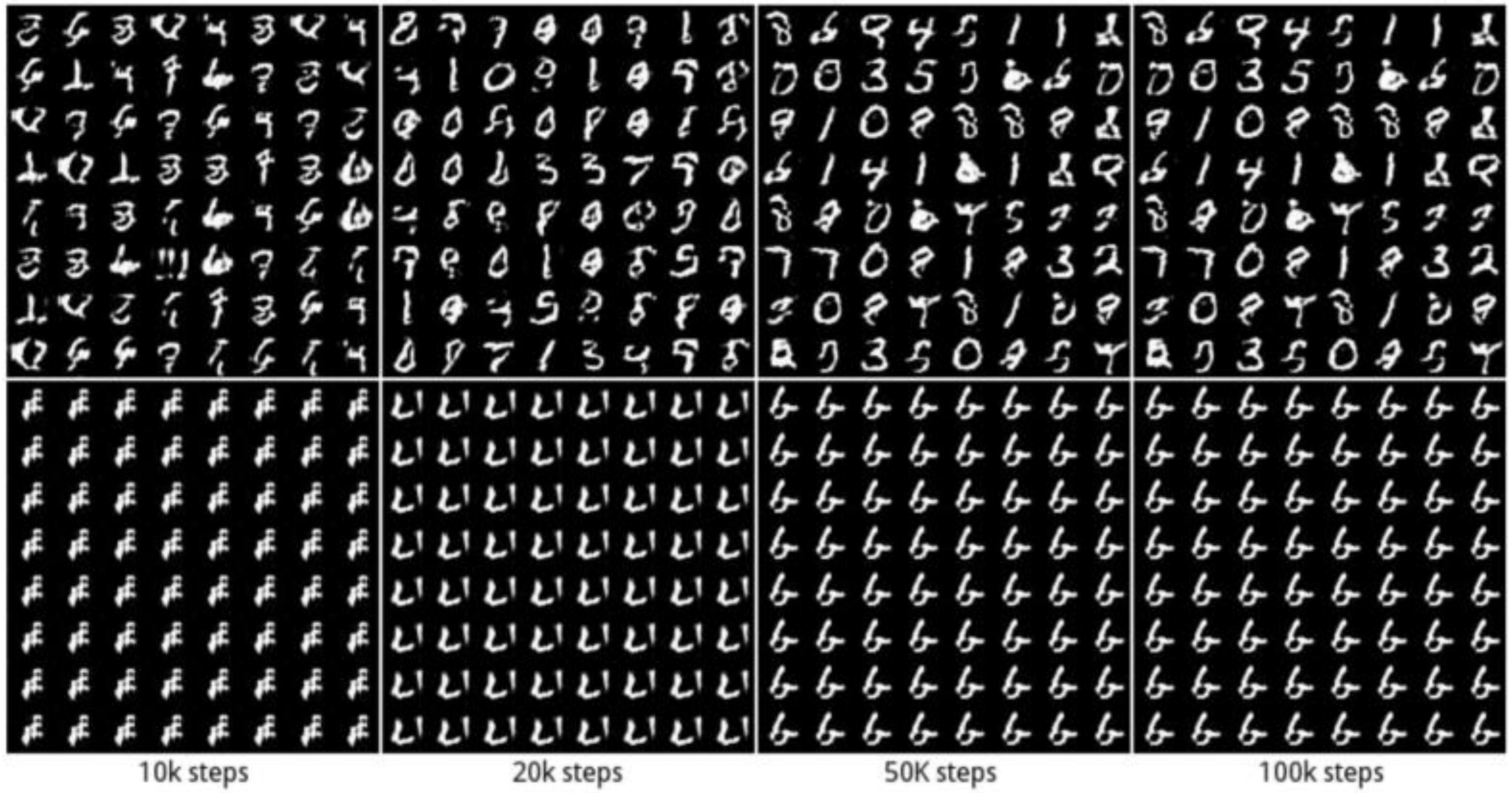
Autoregressive models – pixelCNN 등. Sharp image 생성이 가능하지만, 매우 느리고
latent representation이 없어 applicability에 제한적임.

VAE – 학습이 쉽지만 blurry result.

GAN – Sharp image 생성이 가능하지만, small resolution 에서만 잘 동작하며 training 과정이 unstable.

Problems of training GANs

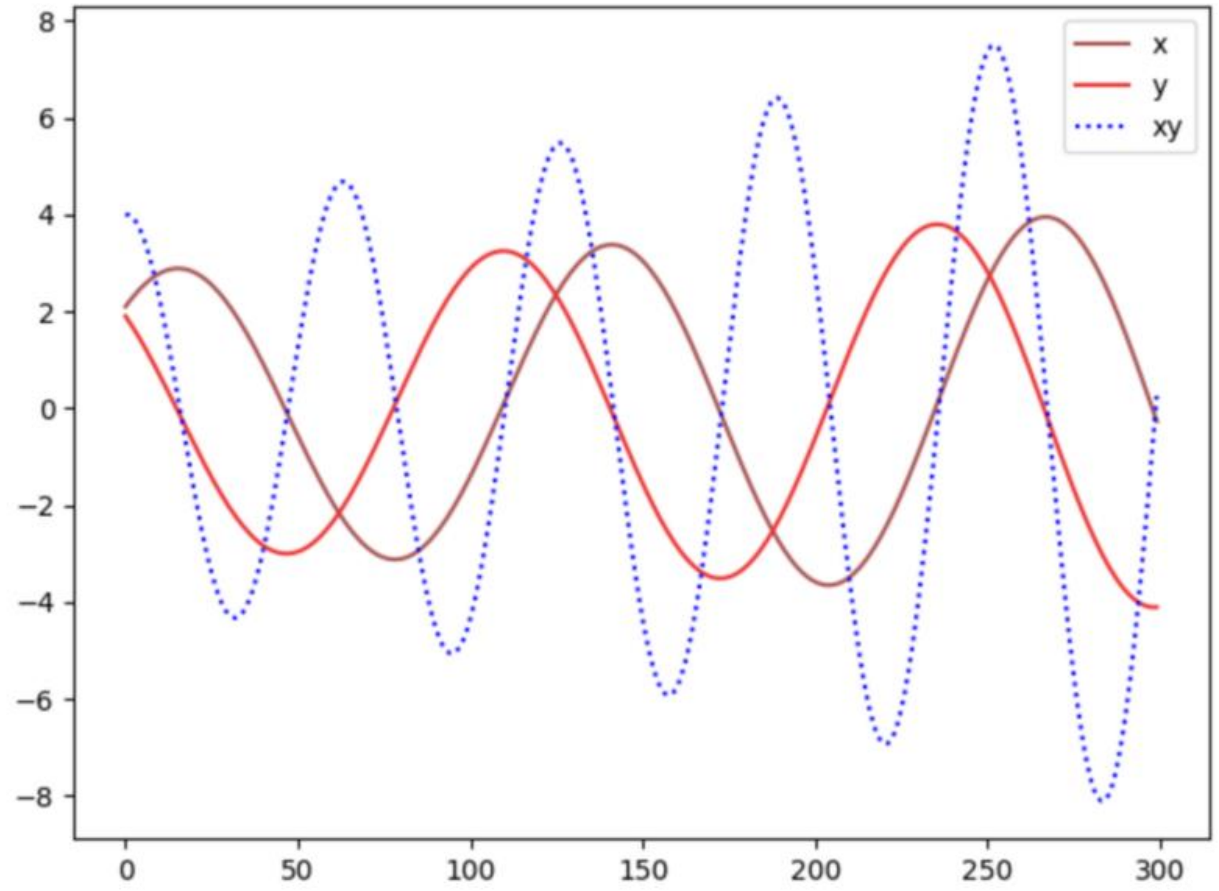
Mode collapse: few 'modes' of data are generated.



Problems of training GANs

Nash Equilibrium: Cost functions may not converge using gradient descent in a minimax game.

$$\min_{B} \max_{A} V(D, G) = xy$$



1. INTRODUCTION

GAN의 문제점을 해결하기 위해 loss function에 대해서 다양한 연구가 있었음.

JS-Divergence, Least-Square, Wasserstein Distance …

해당 논문에서는 loss function 에 대한 연구 방향과는 별개로 새로운 학습 방법론 (training methodology)을 제안함.

2. PROGRESSIVE GROWING OF GANS

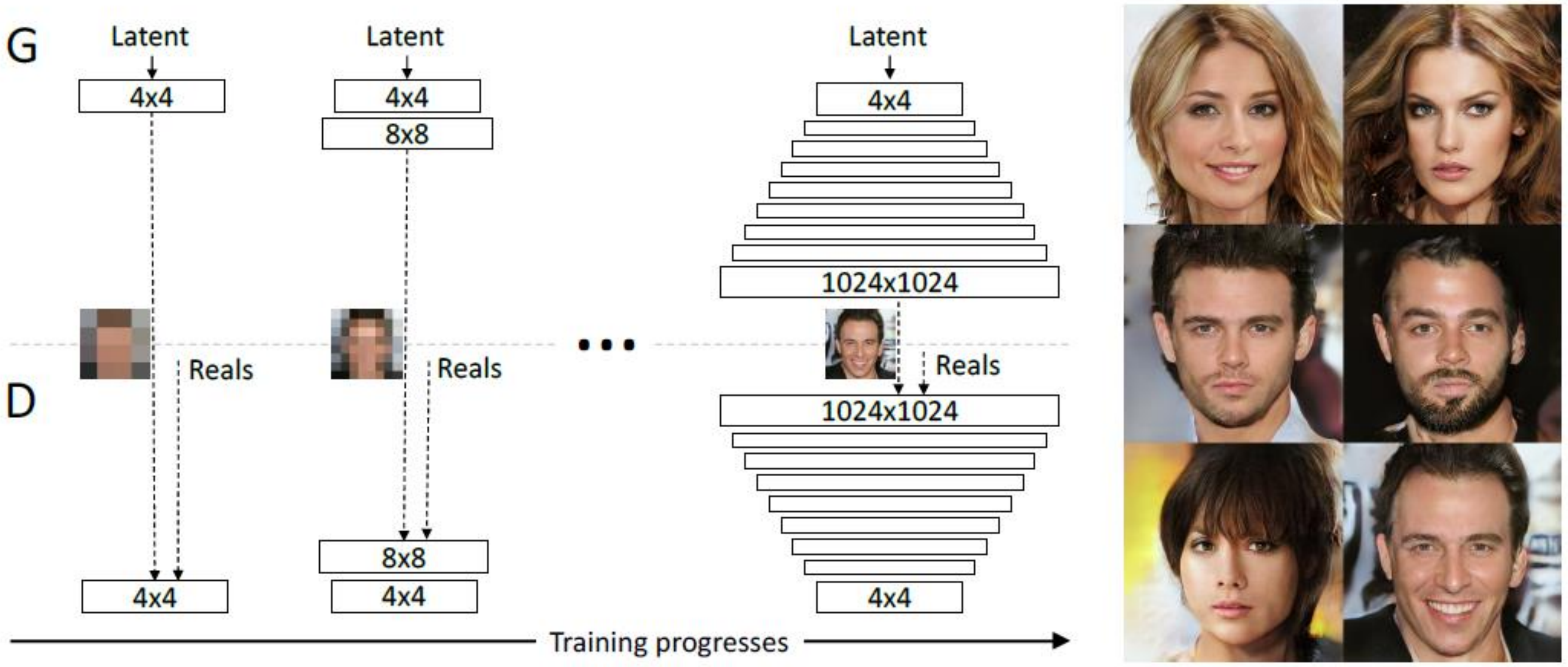


Figure 1: Our training starts with both the generator (G) and discriminator (D) having a low spatial resolution of 4×4 pixels. As the training advances, we incrementally add layers to G and D, thus increasing the spatial resolution of the generated images. All existing layers remain trainable throughout the process. Here $\boxed{N \times N}$ refers to convolutional layers operating on $N \times N$ spatial resolution. This allows stable synthesis in high resolutions and also speeds up training considerably. One the right we show six example images generated using progressive growing at 1024×1024.

2. PROGRESSIVE GROWING OF GANS

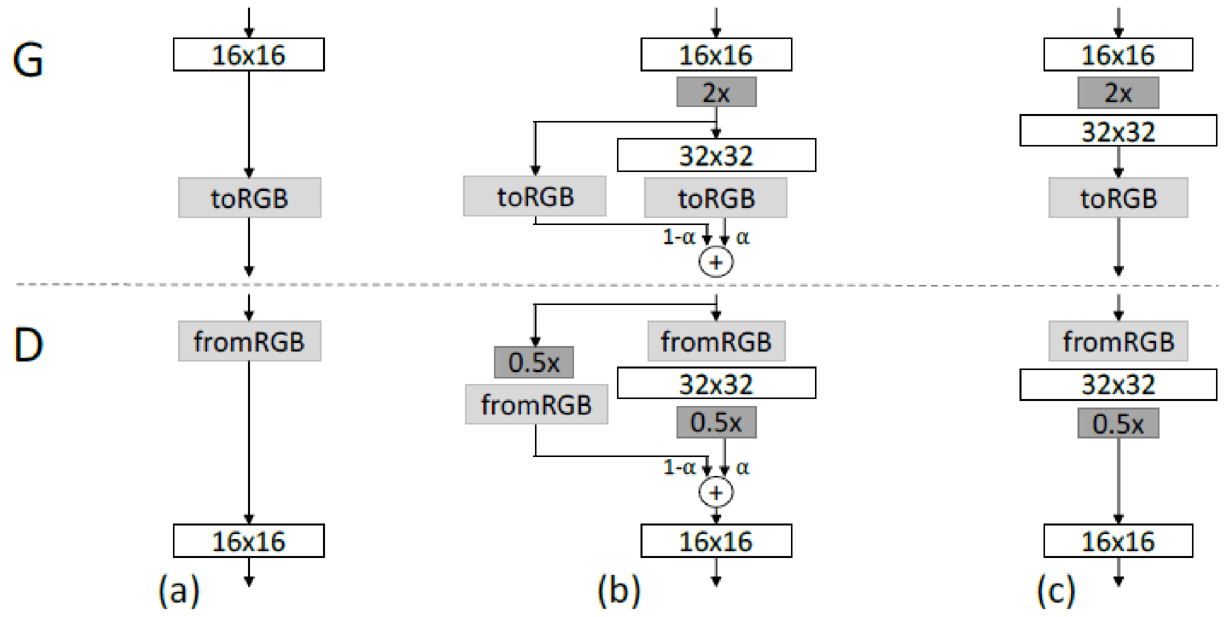


Figure 2: When doubling the resolution of the generator (G) and discriminator (D) we fade in the new layers smoothly. This example illustrates the transition from 16 × 16 images (a) to 32 × 32 images (c). During the transition (b) we treat the layers that operate on the higher resolution like a residual block, whose weight α increases linearly from 0 to 1. Here $\boxed{2\times}$ and $\boxed{0.5\times}$ refer to doubling and halving the image resolution using nearest neighbor filtering and average pooling, respectively. The $\boxed{\text{toRGB}}$ represents a layer that projects feature vectors to RGB colors and $\boxed{\text{fromRGB}}$ does the reverse; both use 1 × 1 convolutions. When training the discriminator, we feed in real images that are downscaled to match the current resolution of the network. During a resolution transition, we interpolate between two resolutions of the real images, similarly to how the generator output combines two resolutions.

3. INCREASING VARIATION USING MINIBATCH STANDARD DEVIATION

Minibatch discrimination 개념을 응용함.

Minibatch discrimination : Discriminator가 batch set 안의 data 들에 대해서 서로 얼마나 유사한지를 판별, generator가 특정한 것만 생성한다고 판단되면 loss 값에 penalty 가 추가됨.

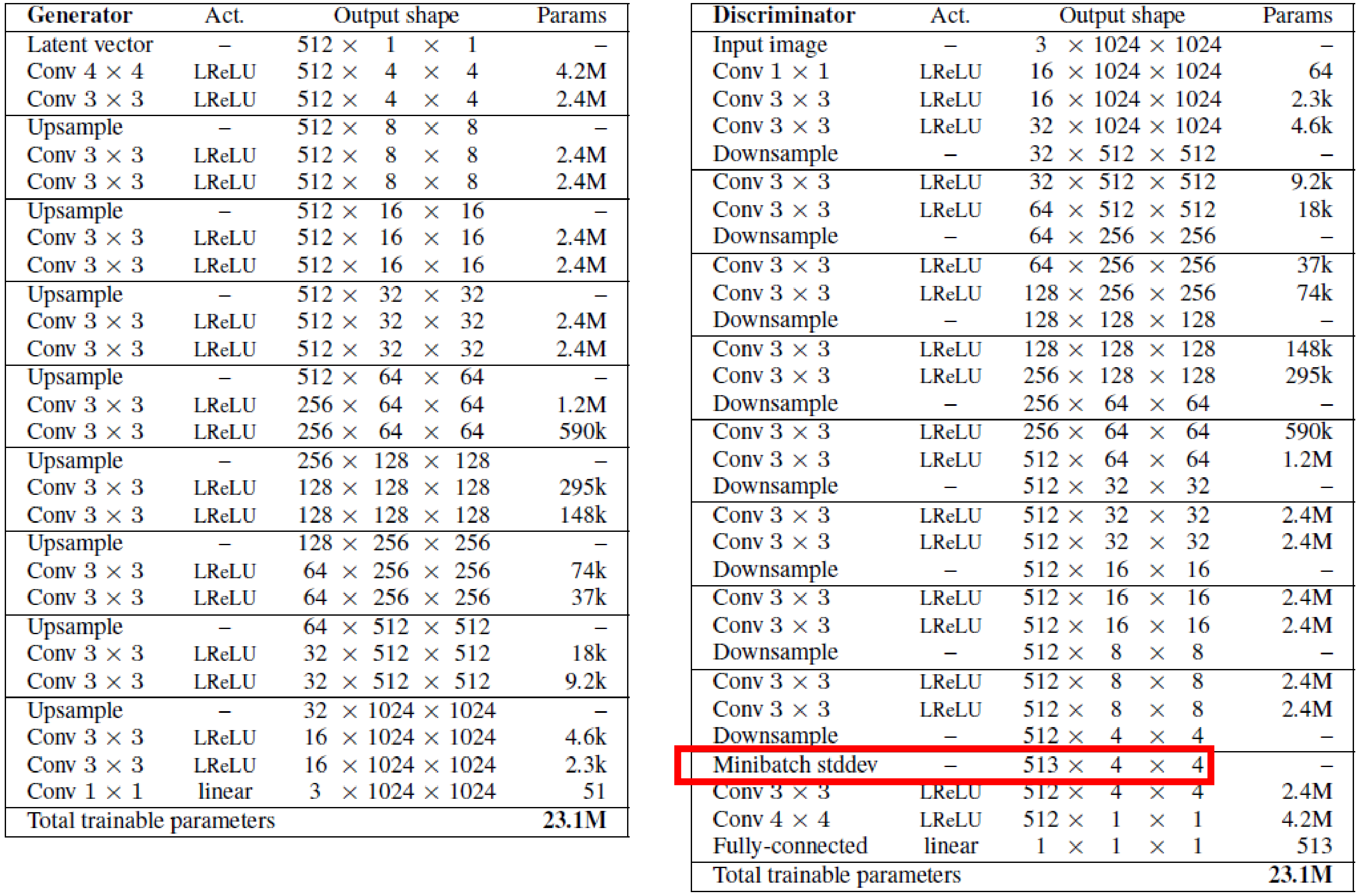
➔ Improve variation of generator

3. INCREASING VARIATION USING MINIBATCH STANDARD DEVIATION

1. Compute the standard deviation for each feature in each spatial location over the minibatch.

2. Average these estimates over all features and spatial locations to arrive at a single value.

3. Replicate the value and concatenate it to all spatial locations and over the minibatch, yielding one additional (constant) feature map.



Generator	Act.	Output shape	Params
Latent vector	–	512 × 1 × 1	–
Conv 4 × 4	LReLU	512 × 4 × 4	4.2M
Conv 3 × 3	LReLU	512 × 4 × 4	2.4M
Upsample	–	512 × 8 × 8	–
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Upsample	–	512 × 16 × 16	–
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Upsample	–	512 × 32 × 32	–
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Upsample	–	512 × 64 × 64	–
Conv 3 × 3	LReLU	256 × 64 × 64	1.2M
Conv 3 × 3	LReLU	256 × 64 × 64	590k
Upsample	–	256 × 128 × 128	–
Conv 3 × 3	LReLU	128 × 128 × 128	295k
Conv 3 × 3	LReLU	128 × 128 × 128	148k
Upsample	–	128 × 256 × 256	–
Conv 3 × 3	LReLU	64 × 256 × 256	74k
Conv 3 × 3	LReLU	64 × 256 × 256	37k
Upsample	–	64 × 512 × 512	–
Conv 3 × 3	LReLU	32 × 512 × 512	18k
Conv 3 × 3	LReLU	32 × 512 × 512	9.2k
Upsample	–	32 × 1024 × 1024	–
Conv 3 × 3	LReLU	16 × 1024 × 1024	4.6k
Conv 3 × 3	LReLU	16 × 1024 × 1024	2.3k
Conv 1 × 1	linear	3 × 1024 × 1024	51
Total trainable parameters			**23.1M**

Discriminator	Act.	Output shape	Params
Input image	–	3 × 1024 × 1024	–
Conv 1 × 1	LReLU	16 × 1024 × 1024	64
Conv 3 × 3	LReLU	16 × 1024 × 1024	2.3k
Conv 3 × 3	LReLU	32 × 1024 × 1024	4.6k
Downsample	–	32 × 512 × 512	–
Conv 3 × 3	LReLU	32 × 512 × 512	9.2k
Conv 3 × 3	LReLU	64 × 512 × 512	18k
Downsample	–	64 × 256 × 256	–
Conv 3 × 3	LReLU	64 × 256 × 256	37k
Conv 3 × 3	LReLU	128 × 256 × 256	74k
Downsample	–	128 × 128 × 128	–
Conv 3 × 3	LReLU	128 × 128 × 128	148k
Conv 3 × 3	LReLU	256 × 128 × 128	295k
Downsample	–	256 × 64 × 64	–
Conv 3 × 3	LReLU	256 × 64 × 64	590k
Conv 3 × 3	LReLU	512 × 64 × 64	1.2M
Downsample	–	512 × 32 × 32	–
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Conv 3 × 3	LReLU	512 × 32 × 32	2.4M
Downsample	–	512 × 16 × 16	–
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Conv 3 × 3	LReLU	512 × 16 × 16	2.4M
Downsample	–	512 × 8 × 8	–
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Conv 3 × 3	LReLU	512 × 8 × 8	2.4M
Downsample	–	512 × 4 × 4	–
Minibatch stddev	–	513 × 4 × 4	–
Conv 3 × 3	LReLU	512 × 4 × 4	2.4M
Conv 4 × 4	LReLU	512 × 1 × 1	4.2M
Fully-connected	linear	1 × 1 × 1	513
Total trainable parameters			**23.1M**

Table 2: Generator and discriminator that we use with CELEBA-HQ to generate 1024×1024 images.

4. NORMALIZATION IN GENERATOR AND DISCRIMINATOR

4.1 Equalized Learning Rate

초기 initialization은 (0, 1) gaussian distribution.

Runtime에서 각 layer마다 He initializer 상수 c ((neuron 개수 / 2) 의 제곱근) 사용해 scaling.

$$\hat{w}_i = w_i / c$$

4.2 Pixelwise Feature Vector Normalization

Local response normalization 개념 사용

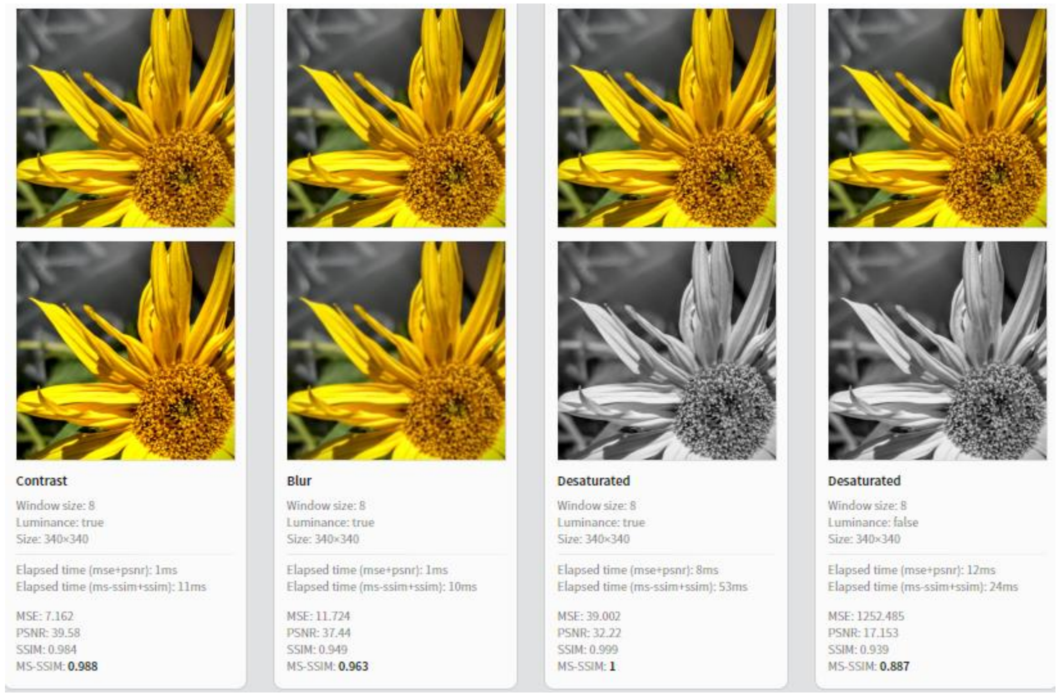
$$b_{x,y} = a_{x,y} / \sqrt{\frac{1}{N}\sum_{j=0}^{N-1}(a_{x,y}^{j})^2 + \epsilon}$$

where $\epsilon = 10^{-8}$, N: number of feature maps, $a_{x,y}$: original feature vector, $b_{x,y}$: normalized feature vector

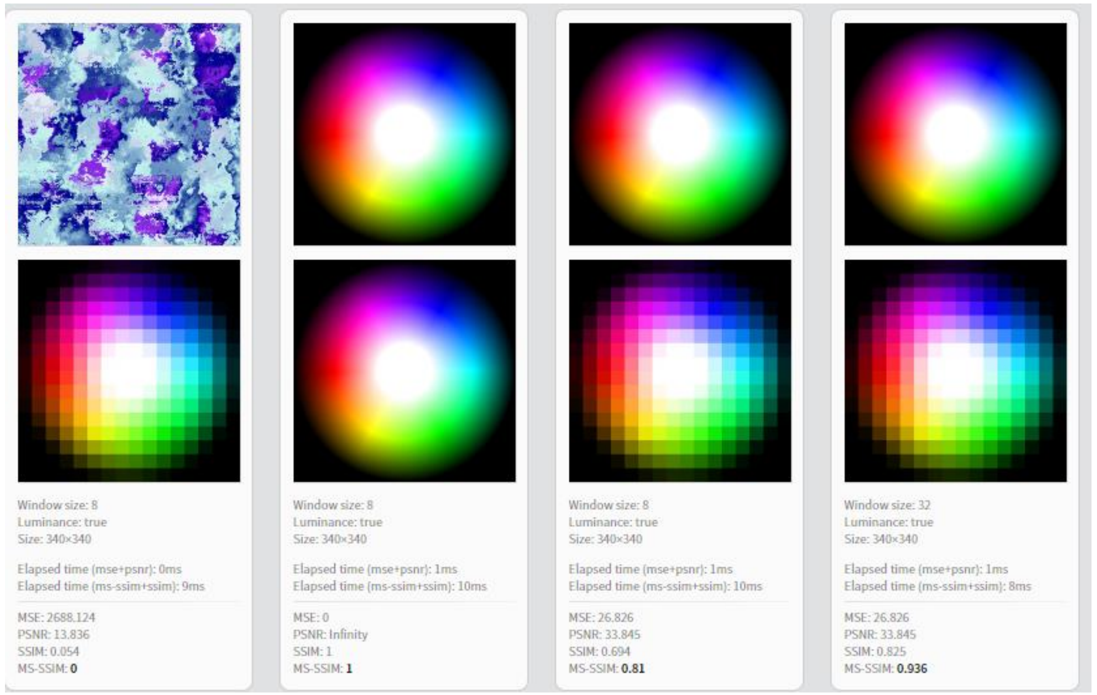
5. MULTI-SCALE STATISTICAL SIMILARITY FOR ASSESSING GAN RESULTS

GAN의 결과가 얼마나 다양하면서 실제와 같은 사진이 나왔는지를 수치로 표현 가능하게 측정한 방법.

기존에 SSIM(Structural Similarity), Multi-Scale SSIM 등이 사용되었음. Large-scale 에서 mode collapse를 잘 찾아냈으나 loss of variations in colors, textures 에 대해서는 잘 측정하지 못함.
이 논문에서는 color, texture 등에 대한 부분까지 보장하기 위해 측정 방식을 약간 수정함.

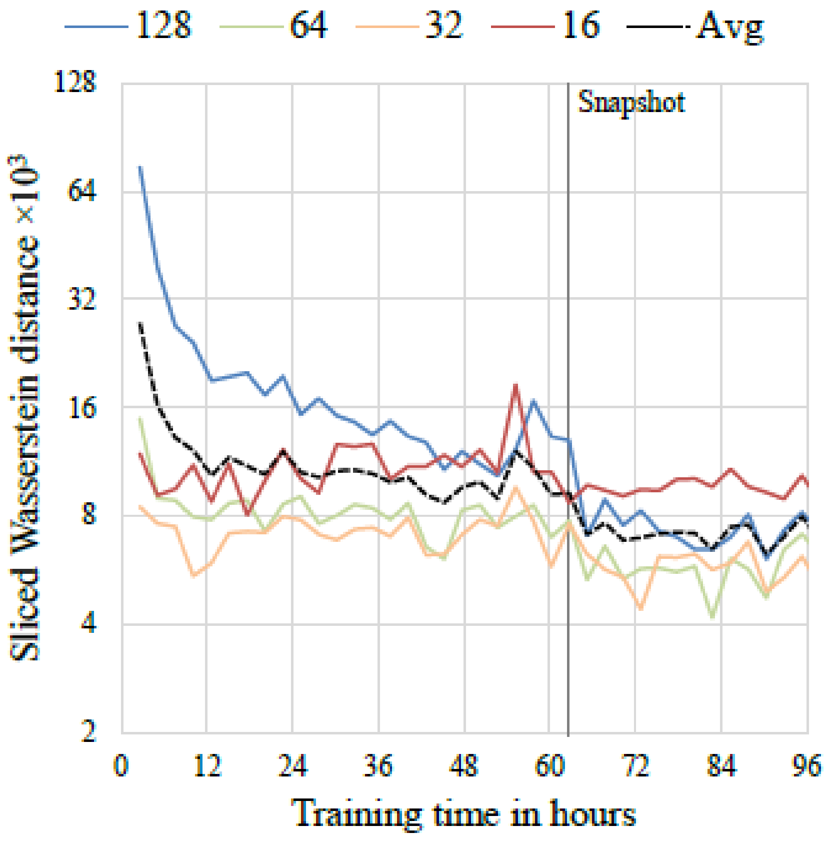


```
Contrast
Window size: 8
Luminance: true
Size: 340×340
Elapsed time (mse+psnr): 1ms
Elapsed time (ms-ssim+ssim): 11ms
MSE: 7.162
PSNR: 39.58
SSIM: 0.984
MS-SSIM: 0.988

Blur
Window size: 8
Luminance: true
Size: 340×340
Elapsed time (mse+psnr): 1ms
Elapsed time (ms-ssim+ssim): 10ms
MSE: 11.724
PSNR: 37.44
SSIM: 0.949
MS-SSIM: 0.963

Desaturated
Window size: 8
Luminance: true
Size: 340×340
Elapsed time (mse+psnr): 8ms
Elapsed time (ms-ssim+ssim): 53ms
MSE: 39.002
PSNR: 32.22
SSIM: 0.999
MS-SSIM: 1

Desaturated
Window size: 8
Luminance: false
Size: 340×340
Elapsed time (mse+psnr): 12ms
Elapsed time (ms-ssim+ssim): 24ms
MSE: 1252.485
PSNR: 17.153
SSIM: 0.939
MS-SSIM: 0.887
```

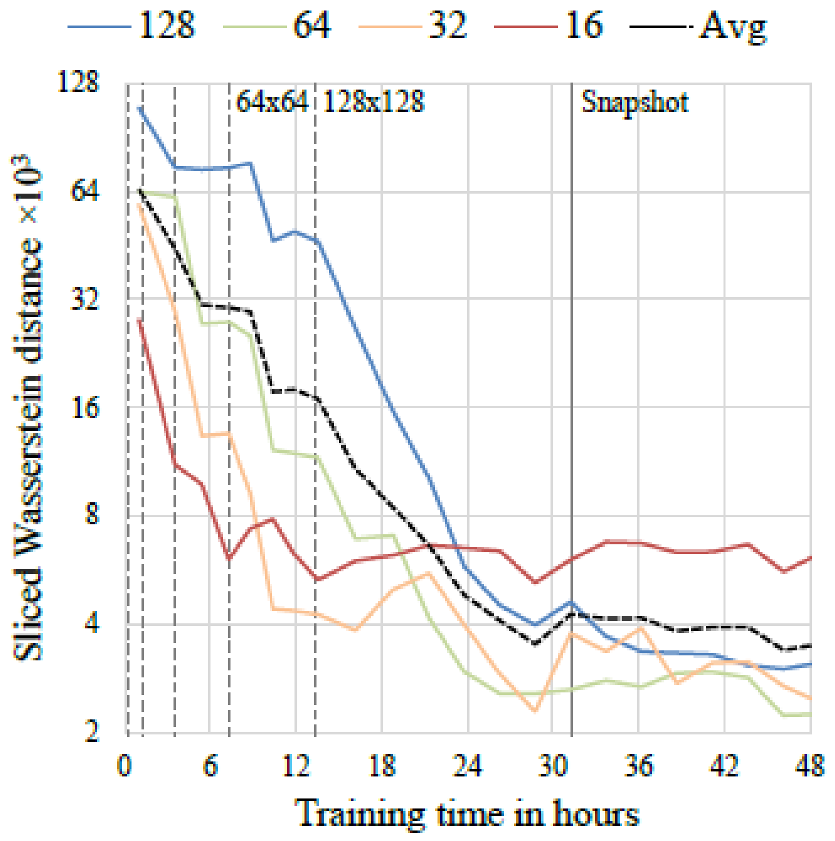


```
Window size: 8
Luminance: true
Size: 340×340
Elapsed time (mse+psnr): 0ms
Elapsed time (ms-ssim+ssim): 9ms
MSE: 2688.124
PSNR: 13.836
SSIM: 0.054
MS-SSIM: 0

Window size: 8
Luminance: true
Size: 340×340
Elapsed time (mse+psnr): 1ms
Elapsed time (ms-ssim+ssim): 10ms
MSE: 0
PSNR: Infinity
SSIM: 1
MS-SSIM: 1

Window size: 8
Luminance: true
Size: 340×340
Elapsed time (mse+psnr): 1ms
Elapsed time (ms-ssim+ssim): 10ms
MSE: 26.826
PSNR: 33.845
SSIM: 0.694
MS-SSIM: 0.81

Window size: 32
Luminance: true
Size: 340×340
Elapsed time (mse+psnr): 1ms
Elapsed time (ms-ssim+ssim): 8ms
MSE: 26.826
PSNR: 33.845
SSIM: 0.825
MS-SSIM: 0.936
```

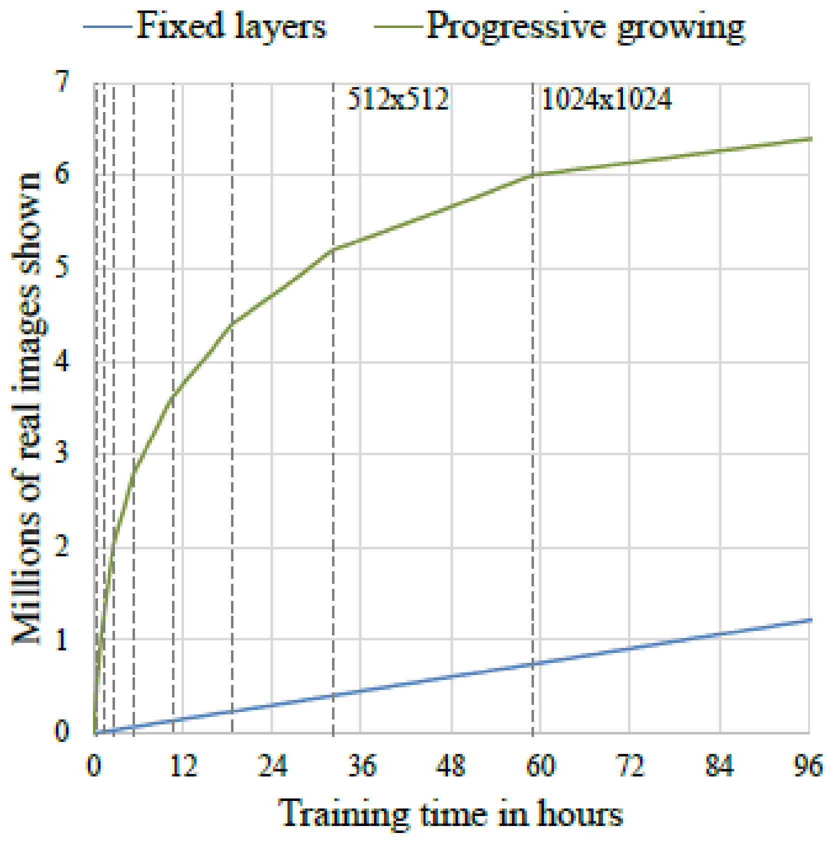
6. EXPERIMENTS



(a)



(b)



(c)

6. EXPERIMENTS

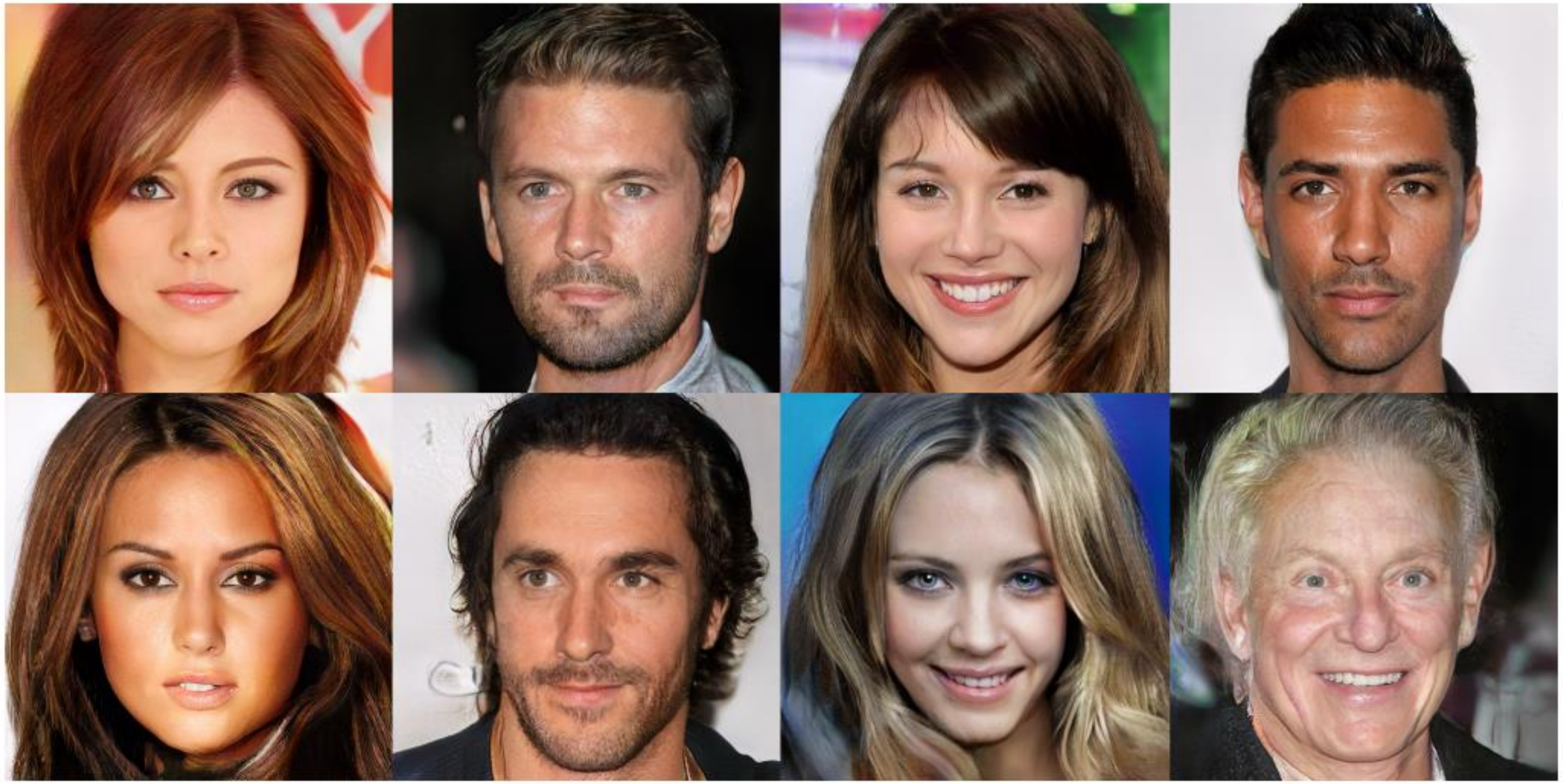


Figure 5: 1024 × 1024 images generated using the CELEBA-HQ dataset. See Appendix F for a larger set of results, and the accompanying video for latent space interpolations.

6. EXPERIMENTS

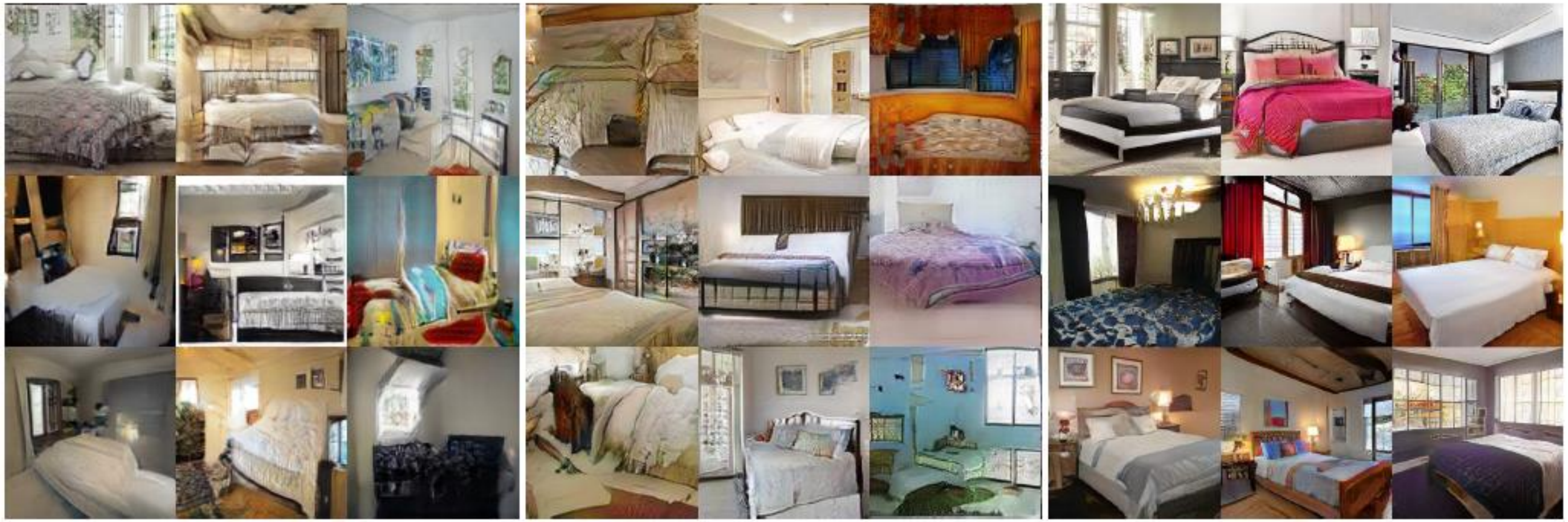

Mao et al. (2016b) (128 × 128) Gulrajani et al. (2017) (128 × 128) Our (256 × 256)

Figure 6: Visual quality comparison in LSUN BEDROOM; pictures copied from the cited articles.

6. EXPERIMENTS

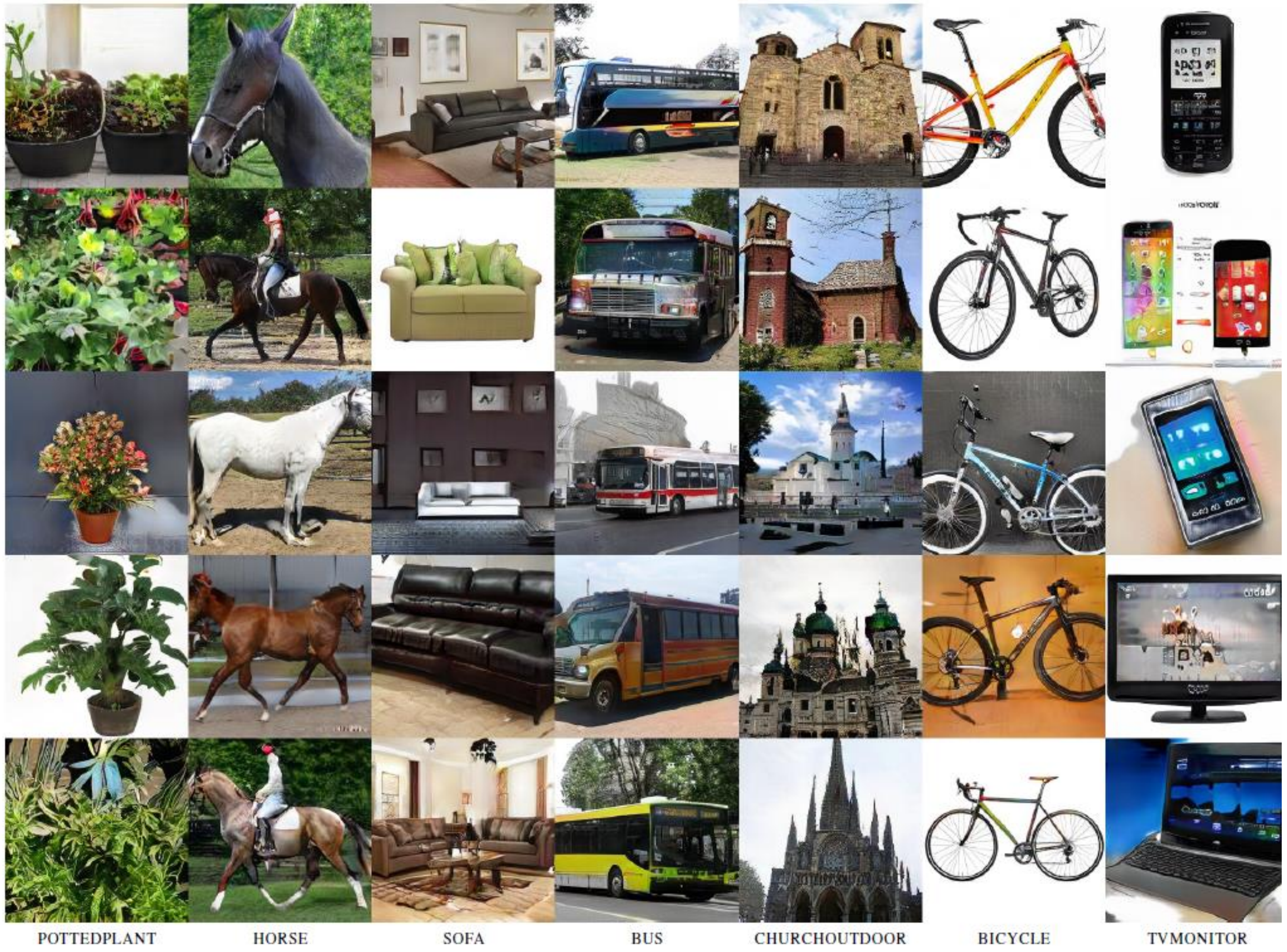


Figure 7: Selection of 256 × 256 images generated from different LSUN categories.

7. Discussions

기존의 GAN 연구와 대비해서 해당 논문이 제안한 방법론이 성능이 가장 좋았다.

Large resolution에 대해서도 고화질의 image를 생성할 수 있었다. (특히 CELEBA-HQ)

학습 방법 + loss function 개선으로 성능을 더 끌어 올릴 수 있을 것으로 생각함.