

A Study on GAN Algorithm for Restoration of Cultural Property (pagoda)

Jin-Hyun Yoon*, Byong-Kwon Lee*, Byung-Wan Kim*

*Professor, Dept. of Multimedia, Seowon University, Chungcheongbuk-do, Korea

*Professor, Dept. of Multimedia, Seowon University, Chungcheongbuk-do, Korea

*Professor, Dept. of Multimedia, Seowon University, Chungcheongbuk-do, Korea

[Abstract]

Today, the restoration of cultural properties is done by applying the latest IT technology from relying on existing data and experts. However, there are cases where new data are released and the original restoration is incorrect. Also, sometimes it takes too long to restore. And there is a possibility that the results will be different than expected. Therefore, we aim to quickly restore cultural properties using DeepLearning. Recently, so the algorithm DcGAN made in GANs algorithm, and image creation, restoring sectors are constantly evolving. We try to find the optimal GAN algorithm for the restoration of cultural properties among various GAN algorithms. Because the GAN algorithm is used in various fields. In the field of restoring cultural properties, it will show that it can be applied in practice by obtaining meaningful results. As a result of experimenting with the DCGAN and Style GAN algorithms among the GAN algorithms, it was confirmed that the DCGAN algorithm generates a top image with a low resolution.

▶ **Key words:** GAN, DCGAN, StyleGAN, TENSORFLOW, PYTORCH, AI

[요 약]

오늘날 문화재 복원은, 기존 자료와 전문가에 의존하는 것과 더불어 최신 IT 기술을 적용하여 복원하고 있다. 하지만 새로운 자료가 나와서 기존 복원이 틀리게 되는 경우, 복원하는데 너무 오랜 시간이 걸릴 때도 있다. 그리고 예상과 다른 결과가 나올 가능성도 있다. 이에 우리는 중요 문화재의 복원을 인공지능을 이용하여 빠르게 복원을 해 보고자 한다. 최근에 Generative Adversarial Networks(GANs) 알고리즘에서 DcGAN[2] 알고리즘이 나오면서 이미지 생성, 복원 분야가 지속해서 발전하고 있다. 이에 본 연구에서는 다양한 GAN 알고리즘을 문화재 복원에 GAN 알고리즘을 적용해 보았다. DcGAN과 StyleGAN을 적용하였으며, 유의미한 결과를 얻었다. GAN 알고리즘 중 DCGAN과 Style GAN 알고리즘을 실험한 결과 DCGAN 알고리즘은 학습이 진행되었으며, 낮은 해상도로 탑 이미지가 생성되는 것을 확인했다. 그리고 Style GAN 알고리즘에서도 역시 학습이 진행 되었으며, 탑 이미지가 생성되었다. 결론적으로 GAN 알고리즘을 사용하여 높은 해상도의 탑 이미지를 구할 수 있게 되었다.

▶ **주제어:** GAN, DcGAN, StyleGAN, TENSORFLOW, PYTORCH, AI

-
- First Author: Jin-Hyun Yoon, Corresponding Author: Byong-Kwon Lee
 - *Jin-Hyun Yoon (jh.yoon@kakao.co.kr), Dept. of Multimedia, Seowon University.
 - *Byong-Kwon Lee (sonic747@seowon.ac.kr), Dept. of Multimedia, Seowon University.
 - *Byung-Wan Kim (designercharles@hanmail.net), Dept. of Multimedia, Seowon University.
 - Received: 2020. 11. 26, Revised: 2020. 12. 31, Accepted: 2021. 01. 02.

I. Introduction

오늘날 문화재 복원을 위해서 다양한 AI 알고리즘 방법이 연구되고 있다. 기존 문화재 복원은 전문가의 능력에 의존해서 과거자료 기준으로 복원 및 연구가 진행되었다. 하지만 새로운 자료가 나타나면서 기존 복원이 잘못되는 경우 역시 발생하고 있다. 그리고 복원 시간과 비용이 너무 많이 든다. 이에 우리는 전문가의 노력이나 새로운 자료의 도움 없이 복원을 진행하여, 복원의 결과를 예상해보는 것 역시 필요하다고 생각한다. 최근 인공지능이 발전하면서 이미지 처리 부분 역시 발전하고 있다. 2014년 GAN 알고리즘이 나오면 이미지 복원 분야에서는 새로운 패러다임을 제시가 되었다[1].

본 연구에서는 문화재 복원을 위한 실제 GAN 알고리즘을 실행할 수 있는 개발 환경을 구축하고, GAN 알고리즘을 실행하여 향후 GAN 알고리즘으로 문화재 복원을 할 수 있다는 발전 방향을 모색하고자 한다. 2절에서는 관련 기술에 관한 연구와 3절에서는 GAN 알고리즘 소개와 결과 및 개선 방향을 도출하고, 4절에서는 문제점을 분석하고 향후 발전 방향을 제시하고자 한다.

II. Preliminaries

1. Related works

Generative Adversarial Networks(GANs)은 기존의 딥러닝(Deep learning) 기술에서 활용한 인공신경망과는 다른 학습 방법을 활용하여 사람이 보기에 진짜와 구분하기 힘들 정도로 정교한 가짜 이미지를 만들어내는 기술이다. 기존의 이미지 처리에서 많이 활용되는 딥러닝 기술은 학습데이터에 대하여 다층의 인공신경망 하나를 학습시키는 방법을 활용하였지만, GAN은 2개의 인공신경망의 상호작용을 활용하여[1], 최종적으로 사람이 보기에 진짜인지 구분하기 힘든 가짜 이미지를 만드는 1개의 생성 신경망을 실제 서비스에 활용한다[2][3][4][6].

GAN은 Generator (생성자)와 Discriminator(판별자) 두 개의 모델을 동시에 적대적인 과정으로 학습하는 알고리즘이다. GAN 알고리즘이 나오고 다양한 연구가 되어 Fig.2 같이 다양한 알고리즘이 연구되고 있다. 이러한 알고리즘으로 Fig.1 같이 입력 이미지로부터 새로운 이미지를 만들어 낼 수 있다.

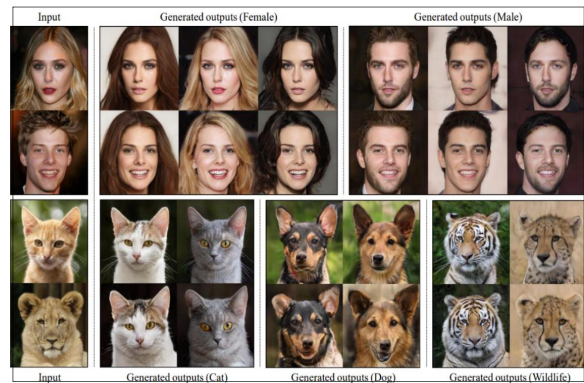


Fig. 1. GAN Result Image

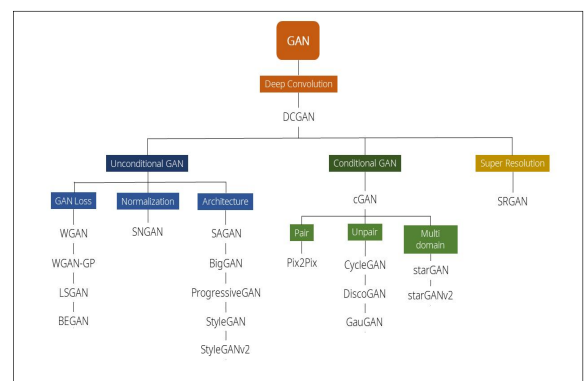


Fig. 2. Direction of GAN development

2.2 GAN Introduce

GAN은 Generator (생성자)와 Discriminator (판별자) 두 개의 모델이 동시에 적대적인 과정으로 학습합니다. 생성자 G는 실제 데이터 분포를 학습하고, 판별자 D는 원래의 데이터인지 생성자로부터 생성이 된 것인지 구분합니다. 생성자 G의 학습 과정은 이미지를 잘 생성해서 속일 확률을 높이고 판별자 D가 제대로 구분하는 확률을 높이는 두 플레이어의 minmax game의 과정이라고 볼 수 있다. Fig. 3은 MINMAX[1] 알고리즘을 수식으로 설명한 것이다.

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

Train jointly in **minimax game** Discriminator outputs likelihood in (0,1) of real image

Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

Discriminator output for real data x
Discriminator output for generated fake data G(z)

- Discriminator (θ_d) wants to **maximize objective** such that $D(x)$ is close to 1 (real) and $D(G(z))$ is close to 0 (fake)
- Generator (θ_g) wants to **minimize objective** such that $D(G(z))$ is close to 1 (discriminator is fooled into thinking generated $G(z)$ is real)

Fig. 3. MinMax algorithm

GAN은 2014년 나온 후에 DCGAN 알고리즘이 나오면서 연구가 활발히 진행되었다. Fig.2를 보면 DCGAN을 시

작으로 최근까지도 지속해서 알고리즘이 연구되고 있다.

연구 방향은 빠르게 객체를 인식해야 하는 분야와는 다르게 해상도를 높여 실제와 같은 이미지를 만드는 방향으로 연구가 진행되고 있다. 이미지를 합성하여 새로운 이미지를 만드는 연구 역시 진행되고 있다.

Fig.2를 보면 GAN은 DCGAN을 시작으로 BEGAN[3], PGGAN[4], BIGGAN[7], starGAN[8]등 다양하고 꾸준하게 연구되고 있다.

2.2.1 DcGAN

기존 GAN에 Convolutional 구조를 적용하였다. 지도 학습(Supervised Learning)에 Convolutional Neural Network (CNN)[1] 을 적용하였으며, 이렇게 학습한 Discriminator가 다른 비지도 학습 알고리즘들과 비등한 수준의 이미지 분류 성능이다. DCGAN[1]에서 성능검증을 위해 Feature Extractor로 실험을 진행하였다. Discriminator는 입력 이미지의 특징을 파악할 수 있는 필터가 학습되는데 특징을 얼마나 잘 학습했는지 알아보는 방법의 하나는 필터들을 이용해 지도학습을 진행한 결과를 보는 것이라고 한다.

2.2.2 BEGAN(Boundary Equilibrium GAN)

EBGAN[12]을 베이스로 하고 Watterstein distance[13]를 사용하였으며, 모델 구조를 단순화하고 이미지 다양성과 품질 간 trade-off를 조절하는 방법 또한 알아냈다고 한다. 학습 방법은 Discriminator를 AutoEncoder[18]로 만들어서, 실제 이미지와 가짜 이미지를 Discriminator에 넣어 pixel-wise reconstruction loss를 구하여, 실제 데이터 분포가 아닌 reconstrution loss를 학습시키는 방법이다. 다음은 그에 따른 loss 분포에 대한내용이다.

$$\mathcal{L}(v) = |v - D(v)|^p \text{ where } \begin{cases} D: \mathbb{R}^{N_x} \mapsto \mathbb{R}^{N_x} & \text{is the autoencoder function.} \\ \eta \in \{1, 2\} & \text{is the target norm.} \\ v \in \mathbb{R}^{N_x} & \text{is a sample of dimension } N_x. \end{cases}$$

η 값이 1이면 L1 loss, 2가 되면 L2 loss가 된다. BEGAN[3]은 L1 loss를 이용했다.

여기서 나온 결과를 이용해 실제 이미지의 loss 분포를 μ_1 , 가짜 이미지의 loss 분포율을 μ_2 라고 하여 Wassertein Distance를 구할 수 있다.

$$W_1(\mu_1, \mu_2) = \inf_{\gamma \in \Gamma(\mu_1, \mu_2)} \mathbb{E}_{(x_1, x_2) \sim \gamma} [|x_1 - x_2|]$$

위의 공식은 Wassertein Distance 공식이고 Distance 의 하한을 구하면 다음과 같다.

$$\inf \mathbb{E}[|x_1 - x_2|] \geq \inf |\mathbb{E}[x_1 - x_2]| = |m_1 - m_2|$$

real data의 loss에 대한 정규분포와, fake data의 loss에 대한 정규분포가 있으면 두 loss 사이에 inf를 만족하는 방법은 a와 b가 있다. BEGAN[3]은 (b)를 선택했는데 Real-Image의 m_1 을 (a)처럼 크게 만들어 버리면 loss가 커지기 때문에 (b)를 선택했다.

$$(a) \begin{cases} W(\mu_1, \mu_2) \propto m_1 - m_2 \\ m_1 \rightarrow \infty \\ m_2 \rightarrow 0 \end{cases} \quad \text{or (b)} \begin{cases} W(\mu_1, \mu_2) \propto m_2 - m_1 \\ m_1 \rightarrow 0 \\ m_2 \rightarrow \infty \end{cases}$$

GAN objective와 같이 generator와 discriminator의 loss들로 식을 나타내면 다음과 같다.

$$\begin{cases} L_D = L(x; \theta_D) - L_D(G(z_D; \theta_G); \theta_D) & \text{for } \theta_D \\ L_G = L(G(z_D; \theta_G); \theta_D) & \text{for } \theta_G \end{cases}$$

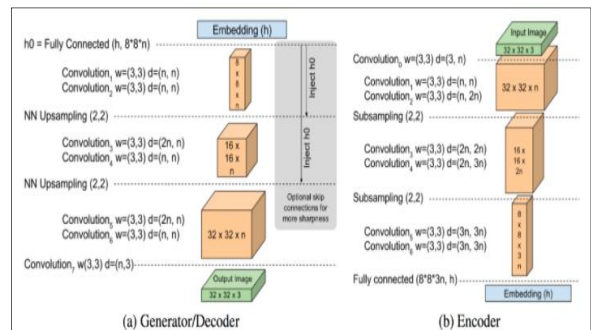


Fig. 4. BEGAN Architecture

Fig. 4를 보면 Discriminator를 대신에 Auto Encoder 를 쓴 것을 알 수 있다.

2.2.3 StyleGAN2

PGGAN[4]을 바탕으로 나온 모델이다. progressive 단계를 적절히 활용할 경우, 이미지의 다양한 시각적 특징을 통제할 수 있다. 저해상도 이미지에서 시작해 단계를 추가해 고해상도 이미지를 점진적으로 생성하는 모델이다. 각 단계에서 입력을 수정해 Table 2.처럼 특징을 조절한다.

Table 1. The performance(FID score) of the model in the different configurations compared to ProGAN. The lower score the better the model.

Method	CelebA-HQ	FFHQ
(A)Baseline Progressive GAN	7.79	8.04
(B)Tuning	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.은 StyleGAN 모델의 방법을 순서대로 설명한 것이다.

Table 2. feature of StyleGAN

Characteristic	Resolution and content
Coarse(Bold features)	(4x4 layer~8x8 layer) Affects poses, general hairstyles, face shapes, etc.
Middle(Medium feature)	(16x16 layer ~ 32x32 layer) 자세한 얼굴 특징, 헤어스타일, 눈 뜨고/Influence on attenuation
Fine(Detailed features)	(64x64 layer ~ 1024x1024 layer) Up to resolution-Influences on color combinations and subtle features such as eyes, hair and skin

Fig. 5는 latent vector Z 를 모델에 바로 넣는 게 아니라 매핑 네트워크를 거쳐서 나온 w 를 사용한다. PGGAN 처럼 바로 latent vector z 가 모델에 들어가면 training data의 probability density를 따라야 해서 머리카락, 눈, 시선 등을 변경하는데 제한이 된다. 그래서 style GAN은 Fig. 5처럼 mapping network를 사용했고, 이로 인해 latent vector 간 상관관계를 줄일 수 있게 되어 업힘을 줄 일 수 있다.

Mapping Network를 이용해서 z 값을 w 값으로 만들었으면, 스타일을 입힐 차례이다. Synthesis Network는 처음 4x4x512 크기의 텐서로 시작해서 1024x1024x3으로 끝나는 8개의 계층으로 구성되어 있다. 해당 계층 끝날 때마다. AGAIN을 적용한다.

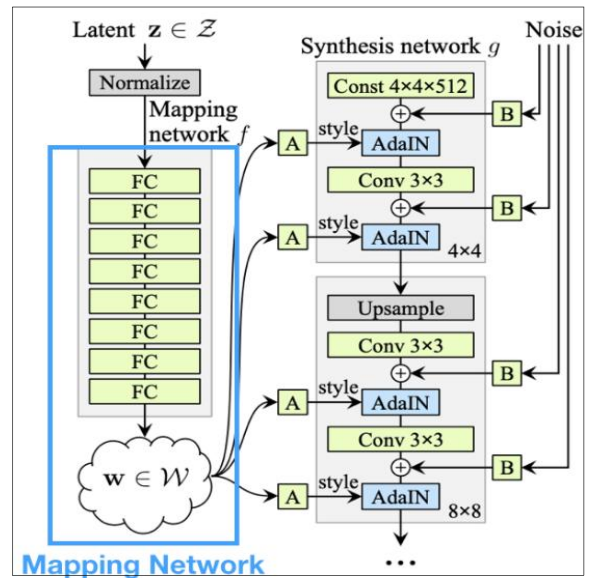


Fig. 5. Style-based generator

Style Modules(AdaIN)는 매핑 네트워크에서 생성된, 인코딩된 정보 w 를 생성된 이미지로 전달(transfer)한다.

III. The Proposed Scheme

3.1 Configuration of test environment

본 연구에서는 앞에서 제시한 DcGAN[1], StyleGAN[2]의 2개의 알고리즘을 선정하여 문화재 복원을 위한 실험을 진행했다. Table 3은 시험을 위한 개발 환경설정이다.

문화재 복원을 위한 인공지능 학습데이터는 학습 수 260개와 학습시간 80시간 이상이 소요됐으며, Fig. 6을 보면 프로그램이 GPU의 CUDA 영역을 사용하여 점유율을 90% 이상 사용하는 것을 알 수 있다.

Table 3. Configuration of test environment

DIV	Configuration
OS	Windows 10
RAM	16GB
Graphics Card	1080ti
Language	Phython3.7.7
platform	tensorflow-gpu 2.3.1 phy-torch 1.5.1
Learning data	AI HUB(Pagoda Image)
Learning time	80 Hour
Image size	1920*1280
Learning Data	150

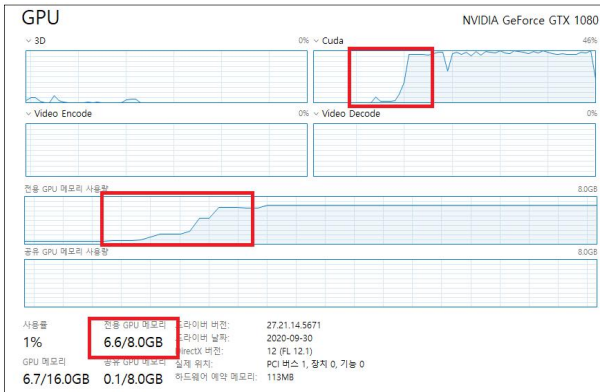


Fig. 6. GPU usage rate for artificial intelligence learning

표본 데이터는 AI hub[5]의 한국형 사물 이미지 ai 데이터 센터에서 구축한 문화재 이미지를 이용하여 복원을 시도하였다. Fig. 7은 ai hub에서 내려받은 훈련 데이터 예시 화면이다.

하지만 1차 작업 결과 탐이 만들어지지 않고 Fig. 8과 같이 숲이 만들어지는 결과를 보였다.



Fig. 7. training data



Fig. 8. Incorrect DcGAN result

이에 Fig. 10의 절차로 객체 인식을 진행하여 Grab cut 으로 배경 이미지를 제거하여 Fig. 9와 같은 배경이미지가 없고 탐만 정확하게 남은 새로운 훈련 데이터를 만들었다.

훈련데이터가 많을수록, 그리고 이미지가 정확할수록 탐을 정확하게 복원할 수 있다. 260개의 데이터로도 훈련이 되고, 복원이 되었지만, 정확한 복원을 위해서는 훈련 데이터를 얻기 위한 연구 역시 필요하다.



Fig. 9. Modified training data



Fig. 10. Procedure for creating training data

3.2 DCGAN

Original GAN에 CNN만 적용하게 되면 만족스러운 결과가 나올 수 없으므로 table 4와 같이 5가지 방법을 적용했다.

Table 4. Architecture guideline for stable Deep Convolutional GANs

1.Remove non-differentiated layers such as the max-pooling layer, and adjust the size of the feature-map using strided convolution and fractional-strided convolution.



2.Add batch normalization.



3.Remove the fully connected hidden layer.



4.Tanh is used as the activation function of the generator's output stage, and Relu is used for the rest.



5.LeakyReLU is used as the discriminator's activation function.

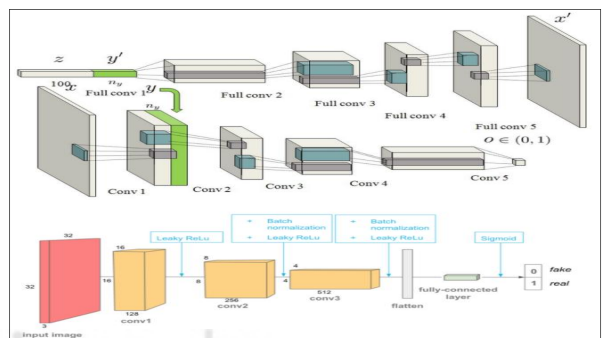


Fig. 11. DCGAN LAYER

```

nn.ConvTranspose2d( nz, ngf * 8, 4, 1, 0, bias=False
),
nn.BatchNorm2d(ngf * 8),
nn.ReLU(True),
nn.ConvTranspose2d(ngf * 8, ngf * 4, 4, 2, 1, bias=
False),
nn.BatchNorm2d(ngf * 4),
nn.ReLU(True),
nn.ConvTranspose2d( ngf * 4, ngf * 2, 4, 2, 1, bias=
False),
nn.BatchNorm2d(ngf * 2),
nn.ReLU(True),
nn.ConvTranspose2d( ngf * 2, ngf, 4, 2, 1, bias=Fals
e),
nn.BatchNorm2d(ngf),
nn.ReLU(True),
nn.ConvTranspose2d( ngf, nc, 4, 2, 1, bias=False),
nn.Tanh()

Generator

nn.Conv2d(nc, ndf, 4, 2, 1, bias=False),
nn.LeakyReLU(0.2, inplace=True),
nn.Conv2d(ndf, ndf * 2, 4, 2, 1, bias=False),
nn.BatchNorm2d(ndf * 2),
nn.LeakyReLU(0.2, inplace=True),
nn.Conv2d(ndf * 2, ndf * 4, 4, 2, 1, bias=False),
nn.BatchNorm2d(ndf * 4),
nn.LeakyReLU(0.2, inplace=True),
nn.Conv2d(ndf * 4, ndf * 8, 4, 2, 1, bias=False),
nn.BatchNorm2d(ndf * 8),
nn.LeakyReLU(0.2, inplace=True),
nn.Conv2d(ndf * 8, 1, 4, 1, 0, bias=False),
nn.Sigmoid()

Discriminator
    
```

Fig. 12. DCGAN Generator and Discriminator Source

Fig. 12의 소스를 보면 Fig.11의 DCGAN Layer 순서대로 실행되는 것을 알 수 있다.

결과는 Fig.14를 보면 DCGAN 알고리즘으로 학습이 되는 것을 알 수 있다. 학습이 완료되면 Fig. 13에서처럼 Generator가 탑을 만들어냈다. 3층 탑과 4층 탑이 구분되는 것을 알 수 있다. 하지만 훈련 데이터 세트를 충분히 만들지 못하여 잡음이 많고 해상도가 낮은 것을 알 수 있다.



Fig. 13. DCGAN Result

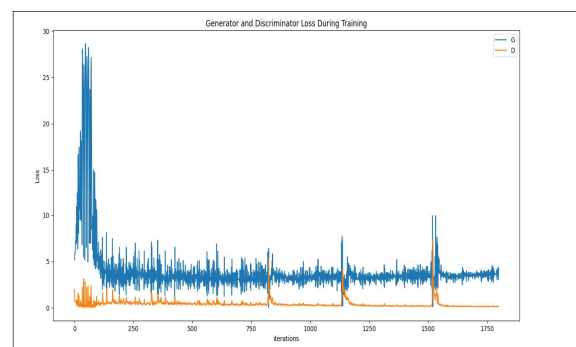


Fig. 14. Generator and Discriminator Loss During Training

3.3 StyleGAN2(Style-based GAN)

PGGAN에서 발전된 것으로 일반 사용자의 작업환경에서도 고해상도의 이미지를 얻을 수 있다.

Style GAN은 PGGAN의 문제점을 해결하는 방법을 제시하였다.

1. 새로운 generator architecture 제안
2. Image 합성 과정에서 scale-specific control 가능
3. Latent space의 interpolation quality 측정 방법 제안
 - Perceptual path length, linear separability

Style GAN의 특징은 input vector가 직접 convolution으로 들어가지 않고 fully connected layer로 구성된 mapping network를 통과하여 w 벡터로 변환된다. 변환된 w 벡터는 constant tensor가 이미지로 변환되는 과정에서 스타일을 입히는 임무를 수행함으로써 다양한 스타일의 이미지를 만들어낼 수 있게 된다.

본 연구의 훈련 데이터가 아닌 Celeba-HQ 이미지로 훈련하였을 때 1 GPU로 약 41일이 걸린다. [2]

GPUs	1024×1024	512×512	256×256
1	41 days 4 hours	24 days 21 hours	14 days 22 hours
2	21 days 22 hours	13 days 7 hours	9 days 5 hours
4	11 days 8 hours	7 days 0 hours	4 days 21 hours
8	6 days 14 hours	4 days 10 hours	3 days 8 hours

Fig. 15. Training times required for images with various resolution compared against no. of GPUs used.

본 연구에서는 훈련 데이터가 크지 않아 오래 걸리지 않으리라고 생각했지만, 3 일차에 메모리 부족으로 중간 결과를 얻을 수 있었다. 결과는 Fig. 16과 같이 탐이 만들어 지다가 중간에 멈춘 것을 알 수 있다.



Fig. 16. StyleGAN Result

IV. Conclusions

본 연구에서 GAN 알고리즘의 학습을 통하여 탐 복원을 시도하였다. 탐을 복원하기 위해서는 정확하고 충분한 훈련 데이터 이미지가 필요하지만 본 연구에서는 충분하지 못한 것을 알 수 있다. 따라서 정확한 훈련 데이터를 만들기 위한 연구 역시 필요하다.

탐 복원을 위해서 GAN 알고리즘 중에 DCGAN과 Style GAN 알고리즘을 실험해 보았다. 실험 결과 학습률중 Loss율이 5% 미만으로 떨어지면서 학습이 되는 것을 알 수 있다. 또한 결과 역시 DCGAN 알고리즘에서는 낮은 해상도로 탐 이미지가 생성되는 것을 확인하였다. 해상도는 낮았지만 3층탑, 4층탑은 구별되게 생성이 되었다. 그리고 Style GAN 알고리즘은 학습이 제대로 진행되었지만, 중간에 멈춰 탐은 구분이 가능하지만, 높은 사양의 PC 성능을 요구하기 때문에 원하는 결과를 얻지 못하였다.

이에 메모리를 제약하는 프로그램을 작성하여 다시 한번 해상도가 높은 GAN 알고리즘을 테스트해 보겠다. 전체

가 안 된다면 이미지를 블록화해서 부분으로 복원하는 방법 역시 연구가 필요하다. 이처럼 GAN알고리즘은 일반 PC에서도 알고리즘을 실행하여 학습과 결과를 확인할 수 있다. 하지만 좋은 성능을 위해서는 시간과 높은 성능의 GPU가 필요하다는 것을 알 수 있었다.

향후 연구과제로 인공지능 학습데이터를 늘리기위해 필요한 정확한 이미지를 자동으로 만들어 주는 프로그램과 시스템 성능을 개선해 학습 속도 개선과 결과의 정확도를 높이는 연구가 진행되어야 할 것으로 사료 된다.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT). (No. 2020R1A2C100766811)

REFERENCES

- [1] Alec Radford, Luke Metz, Soumith Chintala "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", conference paper at ICLR 2016, 2016
- [2] Tero Karras, Samuli Laine, Timo Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4401-4410
- [3] David Berthelot, Thomas Schumm, Luke Metz, "BEGAN: Boundary Equilibrium Generative Adversarial Networks" arXiv:1703.10717, 2017
- [4] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation", arXiv:1710.10196, 2018
- [5] AIHUB, <https://aihub.or.kr>
- [6] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, Jun-Yan Zhu, "Semantic Image Synthesis with Spatially-Adaptive Normalization", arXiv:1903.07291, 2019
- [7] Andrew Brock, Jeff Donahue, Karen Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis", arXiv:1809.11096, 2018
- [8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8789-8797, 2017
- [9] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by

- conditional adversarial autoencoder", In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017
- [10] Mehdi Mirza, Simon Osindero, "Conditional Generative Adversarial Nets", Computer Science, 2014
- [11] Yunjey Choi, Youngjung Uh, Jung-Woo Ha, "StarGAN v2: Diverse Image Synthesis for Multiple Domains", Computer Science 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020
- [12] Junbo Zhao, Michael Mathieu, Yann LeCun, "Energy-based Generative Adversarial Network", arXiv preprint arXiv:1609.03126, 2016.
- [13] Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, Gustavo Rohde, "Generalized Sliced Wasserstein Distances", Advances in Neural Information Processing Systems 32 (NeurIPS 2019), 2019
- [14] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel. "InfoGAN: interpretable representation learning by information maximizing generative adversarial nets.", CoRR, abs/1606.03657, 2016
- [15] V. Dumoulin, J. Shlens, and M. Kudlur. "A learned representation for artistic style.", CoRR, abs/1610.07629, 2016.
- [17] L. A. Gatys, A. S. Ecker, and M. Bethge. "Image style transferring convolutional neural networks.", In Proc. CVPR, 2016
- [18] D. P. Kingma and M. Welling. "Auto-encoding variational bayes.", In ICLR, 2014
- [19] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. "Are GANs created equal? a large-scale study. CoRR", abs/1711.10337, 2017.
- [20] T. Miyato and M. Koyama. "cGANs with projection discriminator.", CoRR, abs/1802.05637, 2018
- [21] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral, "normalization for generative adversarial networks.", CoRR, abs/1802.05957, 2018.
- [22] M. Marchesi. "Megapixel size image creation using generative adversarial networks.", CoRR, abs/1706.00082, 2017

Authors



Jin-Hyun Yoon received the BS degrees in Computer Science from Hansung University, Korea in 2005 and MS degrees in Computer Science from Konkuk University, Korea, in 2013.

MS. Yoon joined the faculty of the Department of Multimedia at Seowon University, Chung-cheong bukdo, Korea, in 2020. He is currently a Professor in the Department of Multimedia, Seowon University. He is interested in IoT, AI, and cloud computing.



Byong-Kwon Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Hanbat, Hannam and Chungbuk University Korea, in 2000, 2003 and 2007, respectively.

My main areas of interest are embedded systems, virtual and augmented reality(VR.AR), and artificial intelligence(AI). The field currently being studied is the construction of an exhibition hall using virtual reality. It is a technology that combines AI with cultural uniform restoration technology as a future research field.



Byung-Wan Kim received the Ph.D. degree in the department of Visual and Multimedia from the Hanyang University in 2012. He is currently a Professor in the Department of Multimedia, Seowon University.

He is interested in UXD/BXD and AR/VR/XR./AI