DCGAN

Deep Convolutional Generative Adversarial Network

YBIGTA 사이언스팀 16기 박준민

목차

Introduction

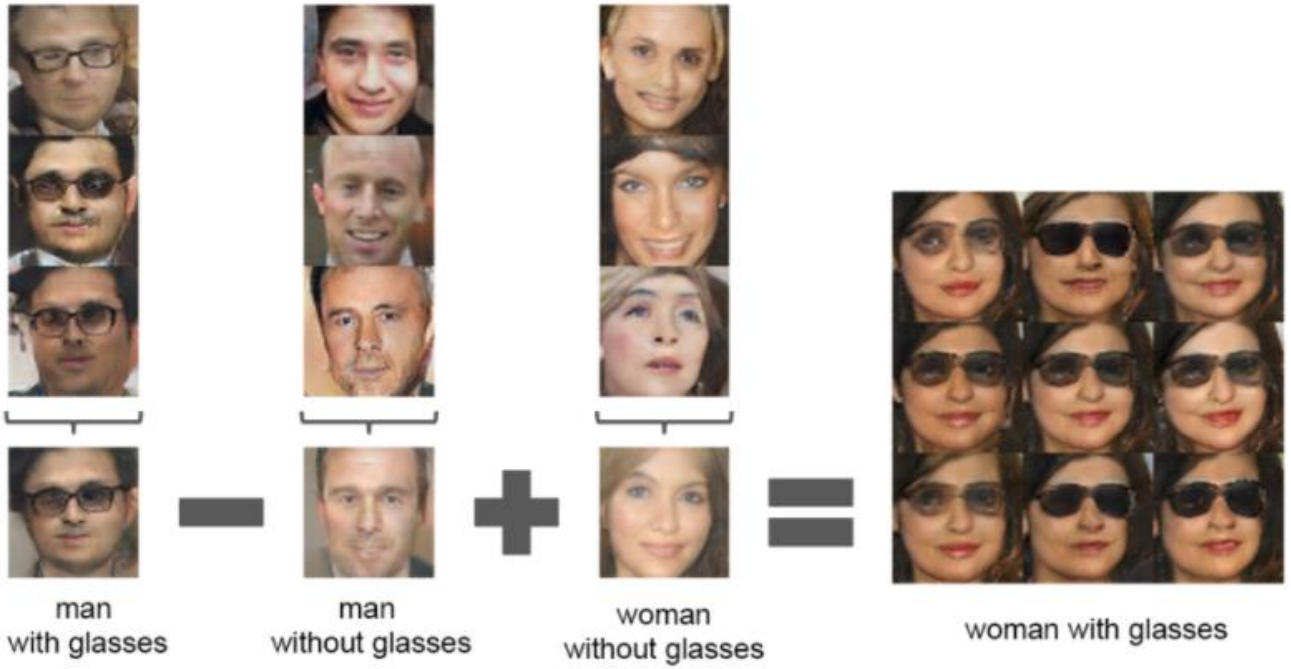
King(왕) – Man(남자) + Woman(여자)

= Queen(여자)

Q) 이미지도 이런 연산이 가능할까?

Introduction

A) 가능하다



By DCGAN

Why DCGAN?

GAN의 한계

1. 결과가 불안정하다

2. Black-box method
   Neural Network의 한계

3. Model Evaluation

Goal of DCGAN

1. Generator가 실제 이미지를 외워서 보여주는 것이 아니다 (Not Overfitting)

2. Generator의 input 공간인 latent sapce(z)에서의 변화가 부드러운 변화로 연결되어야한다.

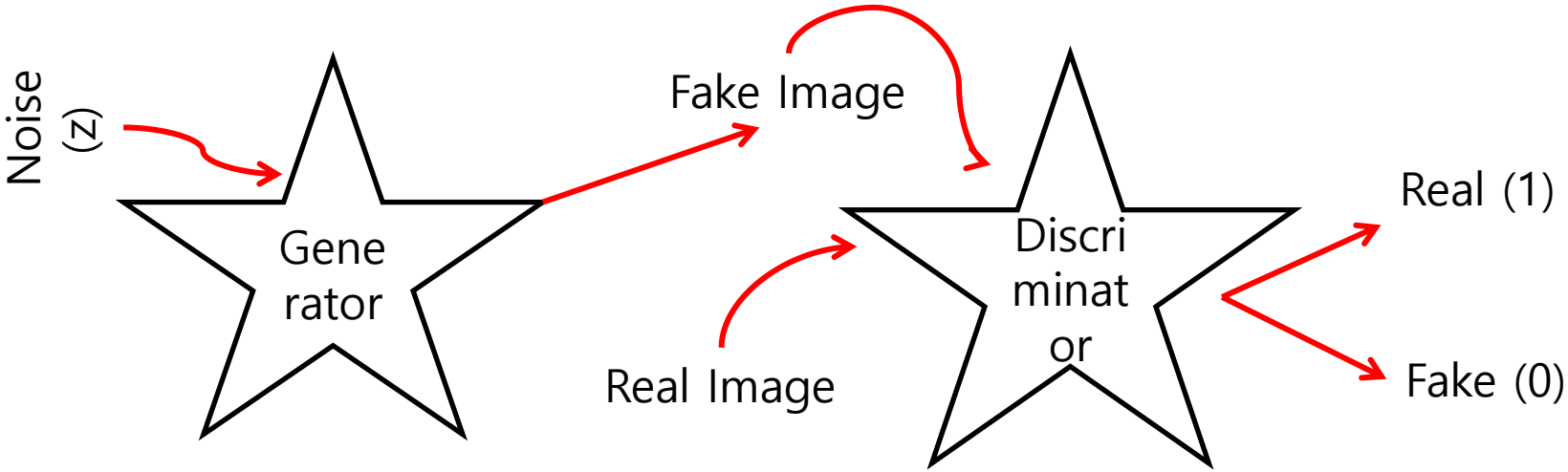
   (walking in the latent space)

Architecture of DCGAN

기본 구조는 GAN과 동일하지만

Generator와 Discriminator의 세부적인 네트워크 구조에서 차이를 보인다

1. 전체적인 구조



Architecture of DCGAN

2. Loss Function

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$$

D : 실제 이미지 x를 받으면 D(x)는 높아지고
가짜 이미지 G(z)를 받으면 D(G(z))는 낮아져야함
=> V(D, G)를 최대화하는 방향으로 학습

G : D(G(z))를 높여야 함
=> V(D, G)를 최소화하는 방향으로 학습

Architecture of DCGAN

3. Generator 구조

1) GAN의 Fully Connected Layer 대신 Transposed Convolutional Layer 사용

2) Batch Normalization 사용

3) Activation Function으로 ReLU 사용하되, output layer 에서는 Tanh 사용

Architecture of DCGAN

3. Generator 구조 (Cont'd)

GAN

```
G = nn.Sequential(
    nn.Linear(d_noise, d_hidden),
    nn.ReLU(),
    nn.Dropout(0.1),
    nn.Linear(d_hidden,d_hidden),
    nn.ReLU(),
    nn.Dropout(0.1),
    nn.Linear(d_hidden, 28*28),
    nn.Tanh()
```

DCGAN

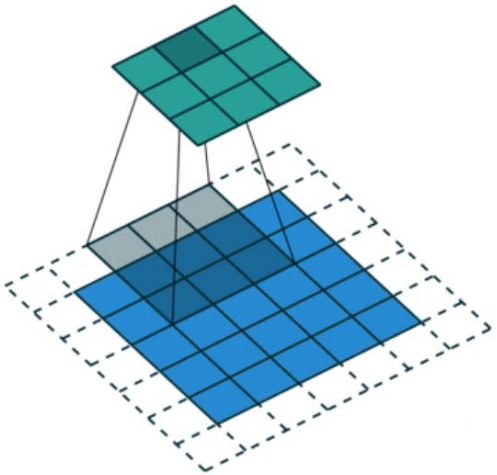
```
class Generator(nn.Module):
    def __init__(self):
        super(Generator, self).__init__()
        self.main = nn.Sequential(
            nn.ConvTranspose2d( nz, ngf * 8, 4, 1, 0, bias=False),
            nn.BatchNorm2d(ngf * 8),
            nn.ReLU(True),
            # (ngf*8) x 4 x 4
            nn.ConvTranspose2d(ngf * 8, ngf * 4, 4, 2, 1, bias=False),
            nn.BatchNorm2d(ngf * 4),
            nn.ReLU(True),
            # (ngf*4) x 8 x 8
            nn.ConvTranspose2d( ngf * 4, ngf * 2, 4, 2, 1, bias=False),
            nn.BatchNorm2d(ngf * 2),
            nn.ReLU(True),
            # (ngf*2) x 16 x 16
            nn.ConvTranspose2d( ngf * 2, ngf, 4, 2, 1, bias=False),
            nn.BatchNorm2d(ngf),
            nn.ReLU(True),
            # (ngf) x 32 x 32
            nn.ConvTranspose2d( ngf, nc, 4, 2, 1, bias=False),
            nn.Tanh()
```

Architecture of DCGAN

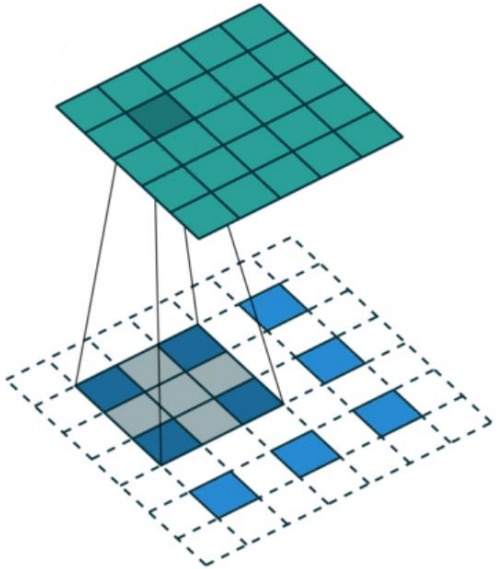
3. Generator 구조 (Cont'd)

Cf) Transposed Convolutional layer란?

Convolutional Layer



Transposed Convolutional Layer



Architecture of DCGAN

4. Discriminator 구조

1) GAN의 Fully Connected Layer 대신 Convolutional Layer 사용

2) Batch Normalization 사용

3) Activation Function으로 Leaky ReLU 사용

Architecture of DCGAN

4. Discriminator 구조 (Cont'd)

GAN

```
D = nn.Sequential(
    nn.Linear(28*28, d_hidden),
    nn.LeakyReLU(),
    nn.Dropout(0.1),
    nn.Linear(d_hidden, d_hidden),
    nn.LeakyReLU(),
    nn.Dropout(0.1),
    nn.Linear(d_hidden, 1),
    nn.Sigmoid()
```

DCGAN

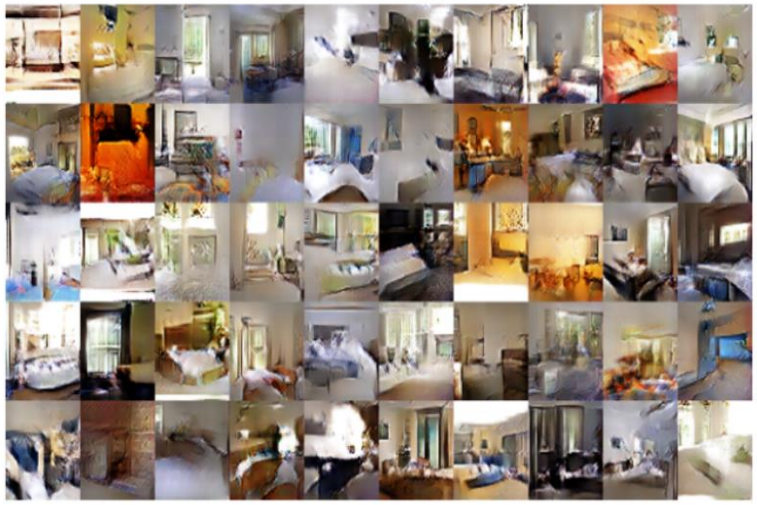
```
class Discriminator(nn.Module):
    def __init__(self):
        super(Discriminator, self).__init__()

        self.main = nn.Sequential(
            # (nc) x 64 x 64
            nn.Conv2d(nc, ndf, 4, 2, 1, bias=False),
            nn.LeakyReLU(0.2, inplace=True),
            # (ndf) x 32 x 32
            nn.Conv2d(ndf, ndf * 2, 4, 2, 1, bias=False),
            nn.BatchNorm2d(ndf * 2),
            nn.LeakyReLU(0.2, inplace=True),
            # (ndf*2) x 16 x 16
            nn.Conv2d(ndf * 2, ndf * 4, 4, 2, 1, bias=False),
            nn.BatchNorm2d(ndf * 4),
            nn.LeakyReLU(0.2, inplace=True),
            # (ndf*4) x 8 x 8
            nn.Conv2d(ndf * 4, ndf * 8, 4, 2, 1, bias=False),
            nn.BatchNorm2d(ndf * 8),
            nn.LeakyReLU(0.2, inplace=True),
            # (ndf*8) x 4 x 4
            nn.Conv2d(ndf * 8, 1, 4, 1, 0, bias=False),
            nn.Sigmoid()
```

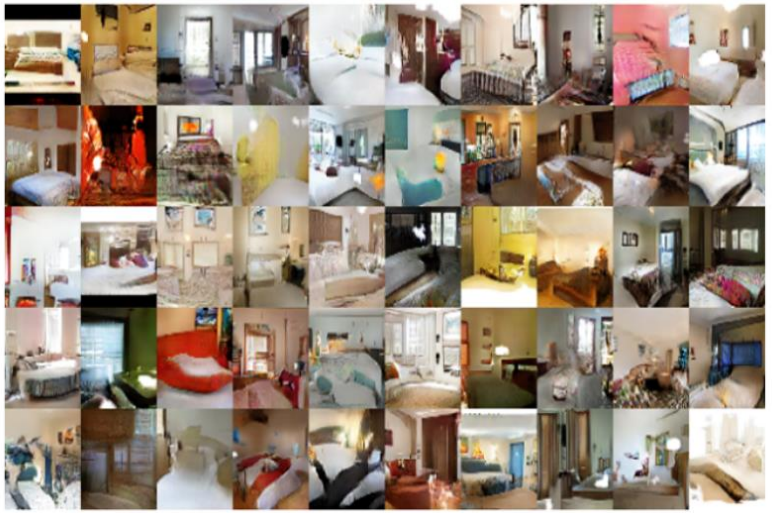
Result of DCGAN

1. Generator가 실제 이미지를 외워서 보여주는 것이 아니다. (Not Overfitting)

1 epoch

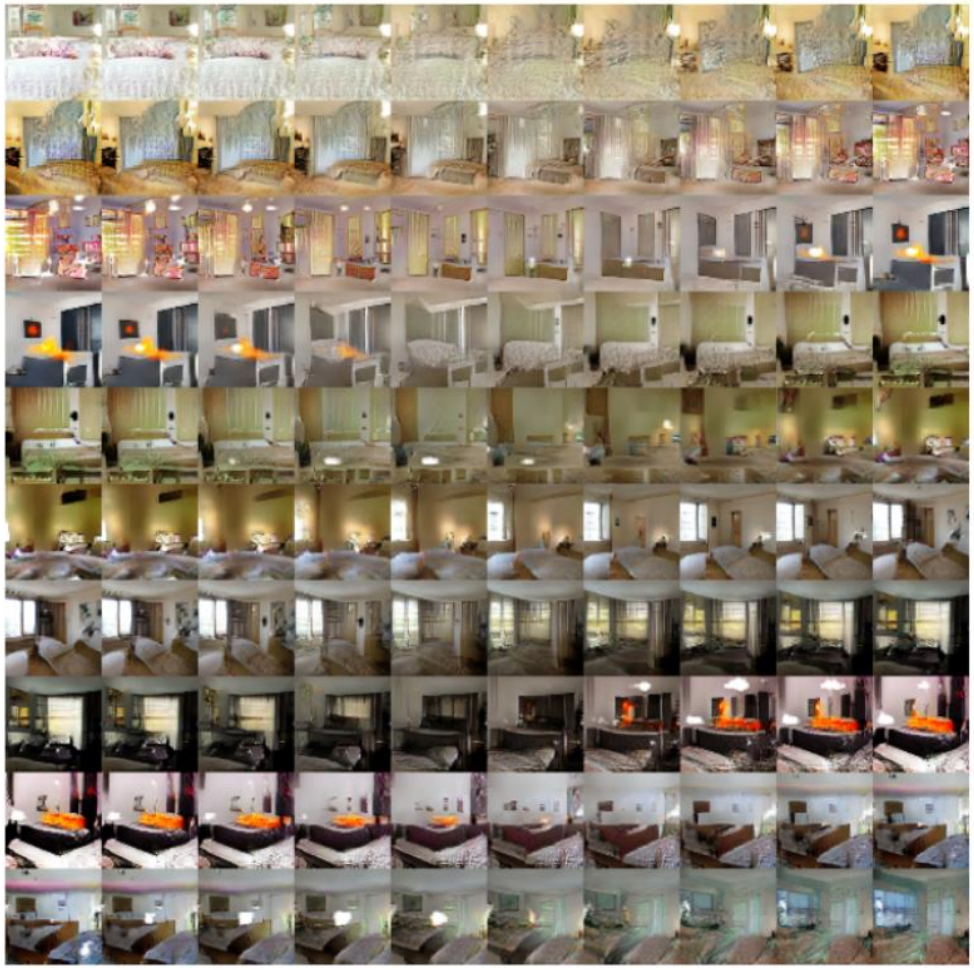


5 epoch



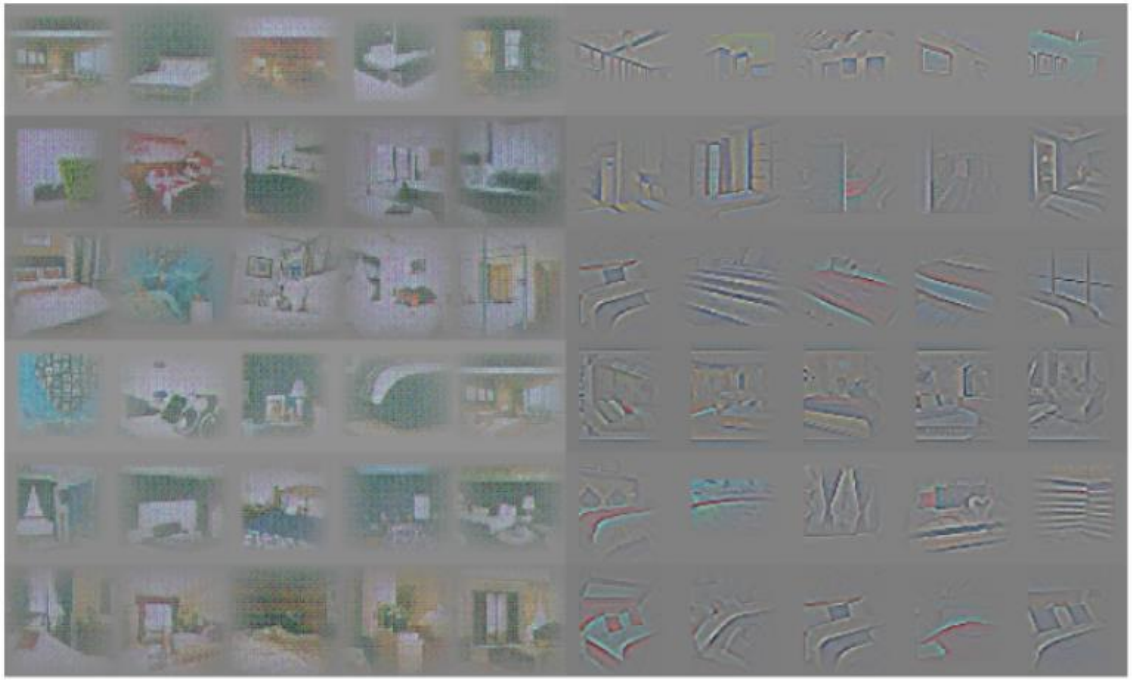
Result of DCGAN

2. Walking in the Latent Space



Result of DCGAN

3. Black-box method 어느정도 해소



Random filters **Trained filters**

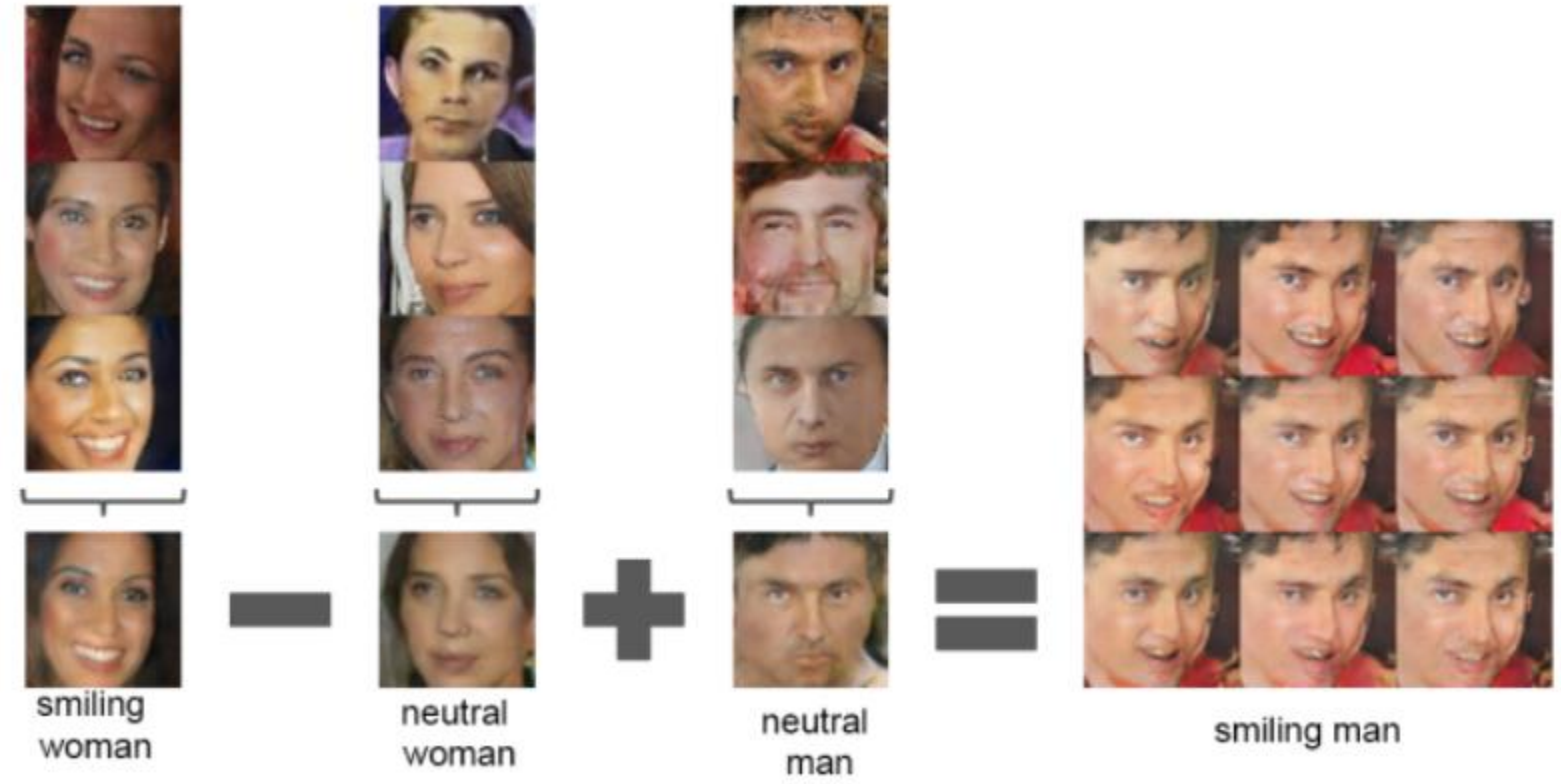
What are the filters looking at?

Result of DCGAN

3. Black-box method 어느정도 해소 (Cont'd)

학습된 특정 filter를 지우면 특징 변화시키기 가능

Result of DCGAN



Reference

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

- 초짜 대학원생의 입장에서 이해하는 Deep Convolutional Generative Adversarial Network (DCGAN) (http://jaejunyoo.blogspot.com/2017/02/deep-convolutional-gan-dcgan-1.html)