

야간 영상을 주간영상으로 바꾸기

CONTENTS

- I. 테스트 결과물 (1)
- II. prerequisite knowledge
- III. GAN 의 개념
- IV. GAN 의 세부 내용
- V. 수식으로서 설명한 GAN
- VI. 코드로서 설명한 GAN
- VII. DCGAN 의 개념
- VIII. PixToPix 의 개념
- IX. Cycle GAN의 개념
- X. 테스트 결과물 (2)
- XI. Cycle GAN 의 특징 정리
- XII. 결론

테스트 결과물 (1)

- 야간 원본 사진



real_B-0



real_B-1



real_B-2



real_B-3



- 주간으로 속성을 변경한 사진



fake_samples-final_BtoA-0



fake_samples-final_BtoA-1



fake_samples-final_BtoA-2



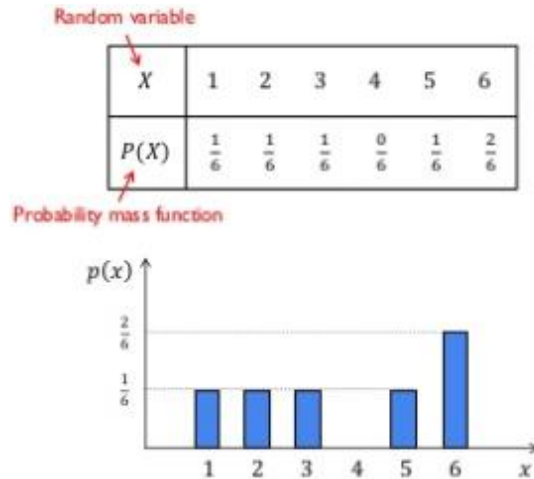
fake_samples-final_BtoA-3



- Cycle GAN (Generative Adversarial Network) 를 활용한 결과

prerequisite knowledge

■ Probability distribution



What if x is actual images in the training data?

At this point, x can be represented as a (for example) $64 \times 64 \times 3$ dimensional vector.

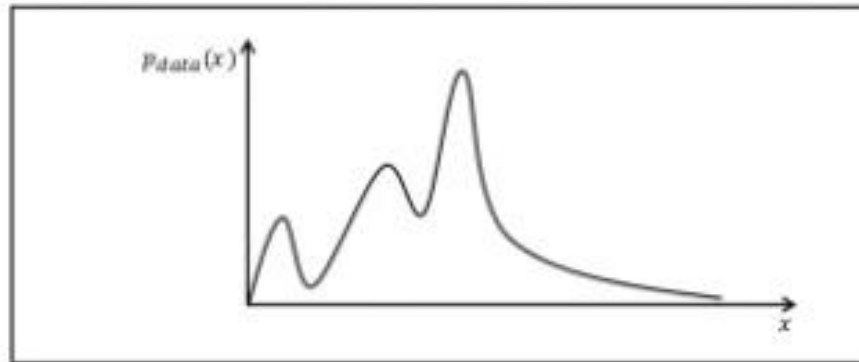


- 어떠한 X 가 존재할 때, 우리는 그에 해당하는 확률 분포 $P(X)$ 가 존재한다고 가정
- $P(X)$ 에 대한 그래프를 나타낼 수 있음

prerequisite knowledge

- Probability distribution

Probability density function
↓
There is a $p_{data}(x)$ that represents the distribution of actual images.

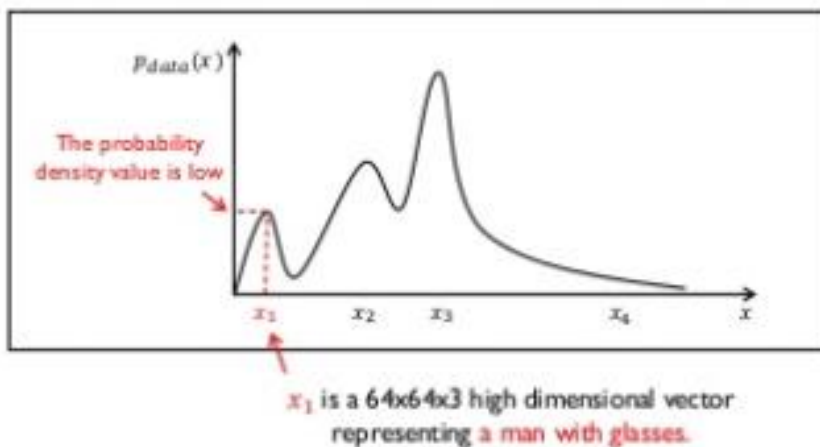


- 우리가 가진 연예인 얼굴 **dataset** 이 위와 같은 분포를 띤다고 가정

prerequisite knowledge

- Probability distribution

Let's take an example with human face image dataset.
Our dataset may contain few images of **men with glasses**.

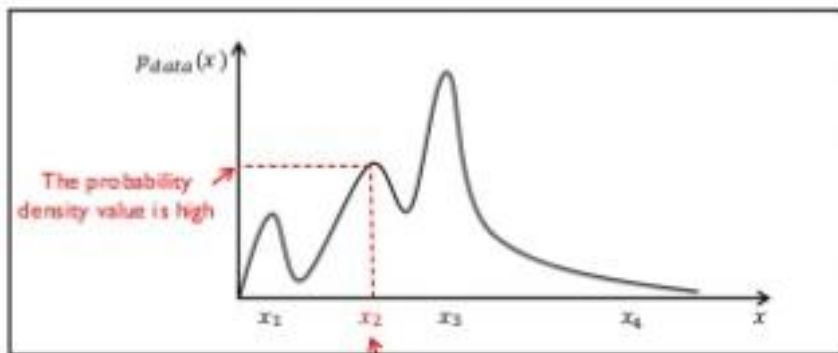


- 확률 분포 내에서 x_1 에 해당하는 것은 안경 쓴 남자들의 얼굴

prerequisite knowledge

- Probability distribution

Our dataset may contain many images of **women with black hair**.



x_2 is a $64 \times 64 \times 3$ high dimensional vector representing **a woman with black hair**.

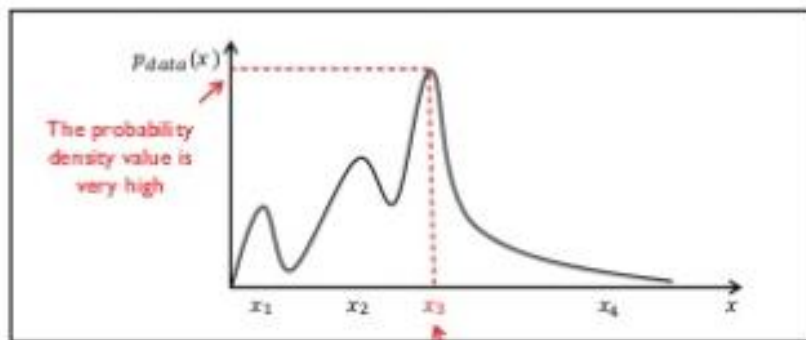


- 확률 분포 내에서 x_2 에 해당하는 것은 흑발의 여성 얼굴

prerequisite knowledge

- Probability distribution

Our dataset may contain very many images of **women with blonde hair**.



x_3 is a $64 \times 64 \times 3$ high dimensional vector representing **a woman with blonde hair**.

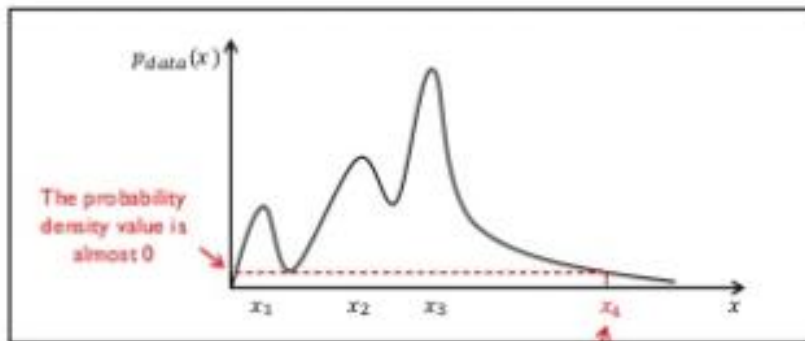


- 확률 분포 내에서 x_3 에 해당하는 것은 금발의 여성 얼굴

prerequisite knowledge

- Probability distribution

Our dataset may not contain these strange images.

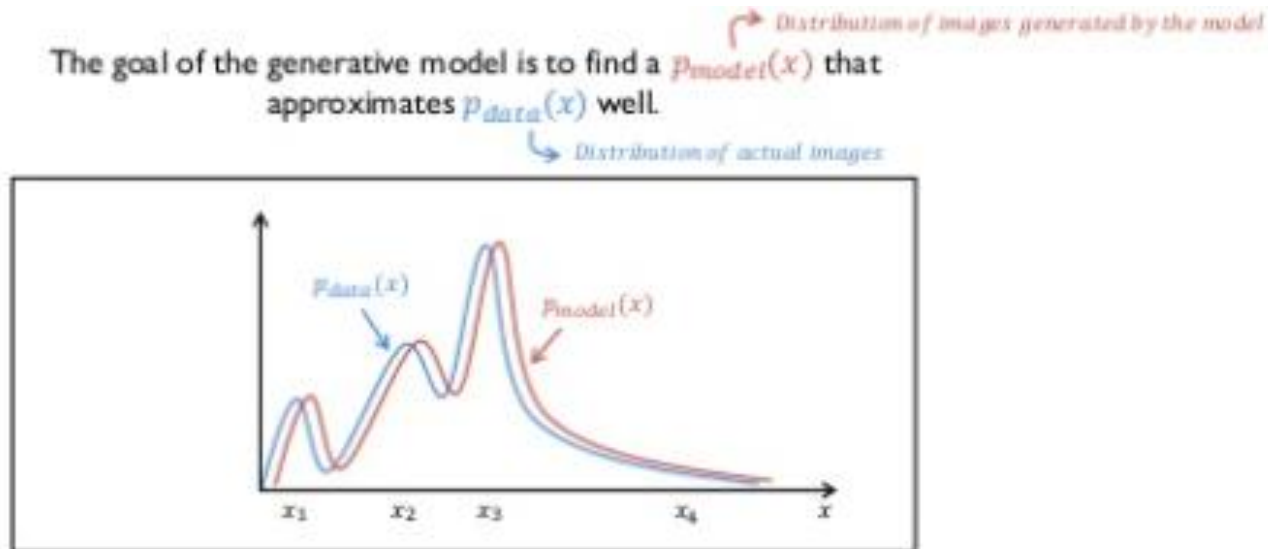


x_4 is an $64 \times 64 \times 3$ high dimensional vector representing very strange images.

- 확률 분포 내에서 예측 값이 거의 0에 수렴하는 x_4 에 해당하는 것은 우리의 데이터 셋에는 존재하지 않는 이상한 얼굴

prerequisite knowledge

- Probability distribution



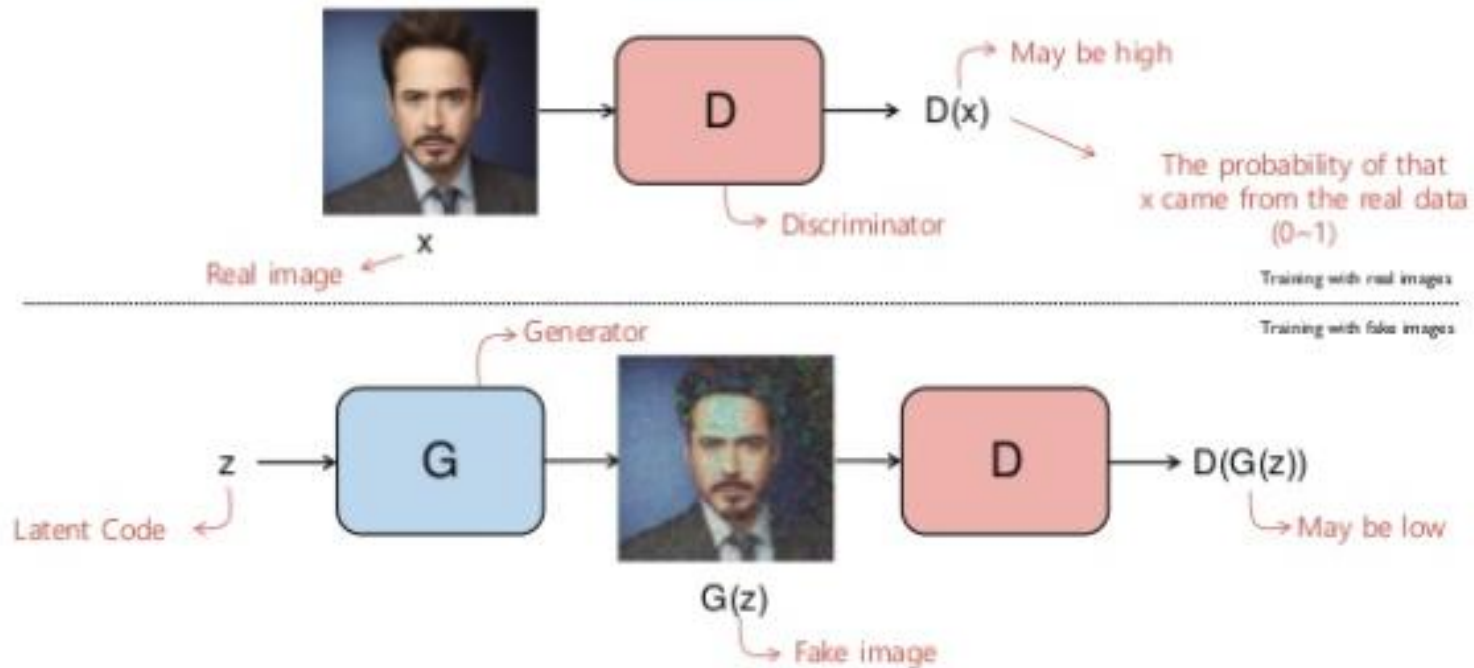
- GAN의 핵심 내용은 우리가 예측하고자 하는 $P_{model}(X)$ 를 실제 데이터 셋이 갖는 확률 분포 그래프인 $P_{data}(X)$ 와 같게 만드는 것
- 그렇게 된다면 특정 X 값 (latent Z code)을 뽑았을 때, 그에 매칭되는 사람 얼굴을 만들어 낼 수 있게 됨

GAN 의 개념



- G (Generator) 는 만드는 녀석
- D (Discriminator) 는 구별하는 녀석

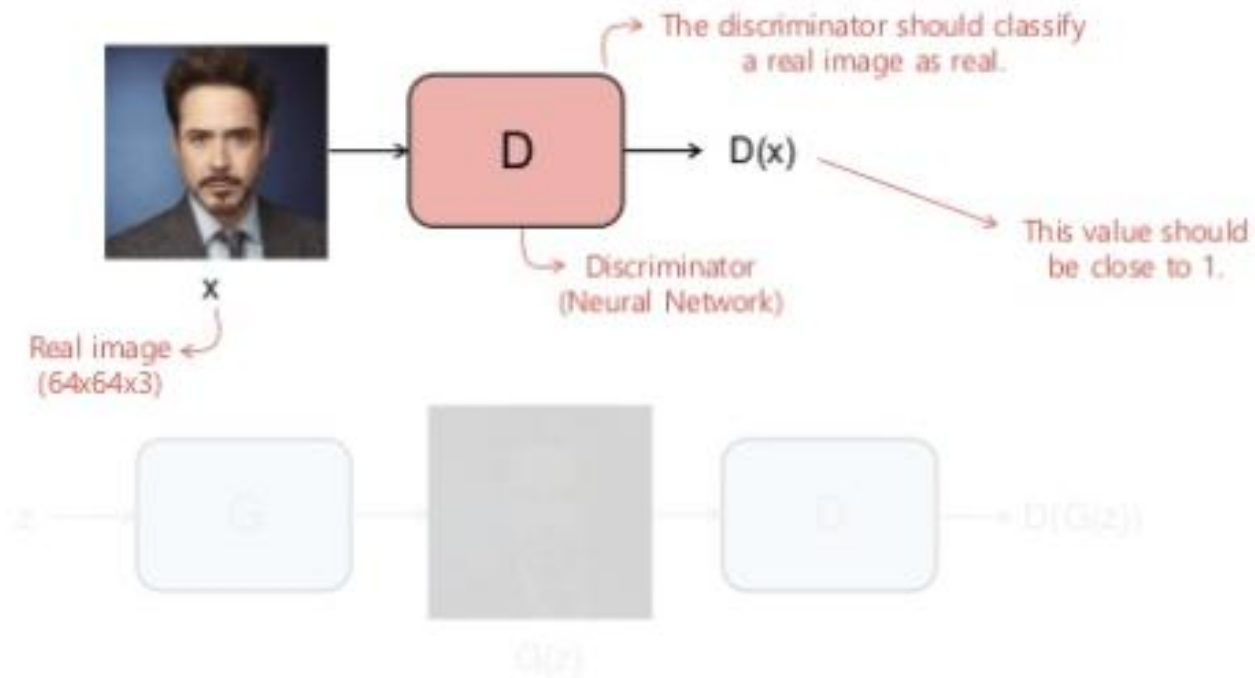
GAN 의 세부 내용



- GAN 은 앞서 설명드린 것과 같이 **G**와 **D** 로 구성 (자세한 이야기는 뒷장..)

GAN의 세부 내용

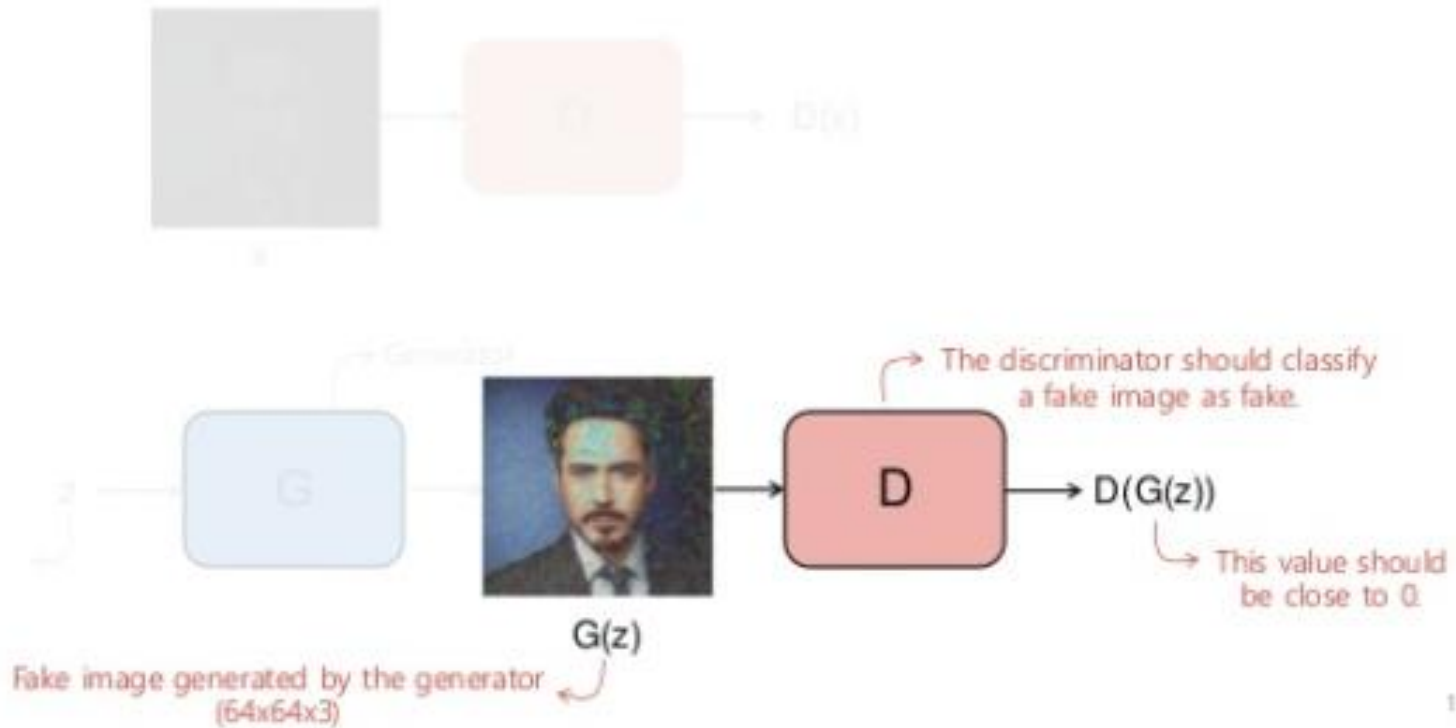
- D의 입장



- D가 하는 일은 입력으로 실제 이미지가 들어올 경우 1,

GAN 의 세부 내용

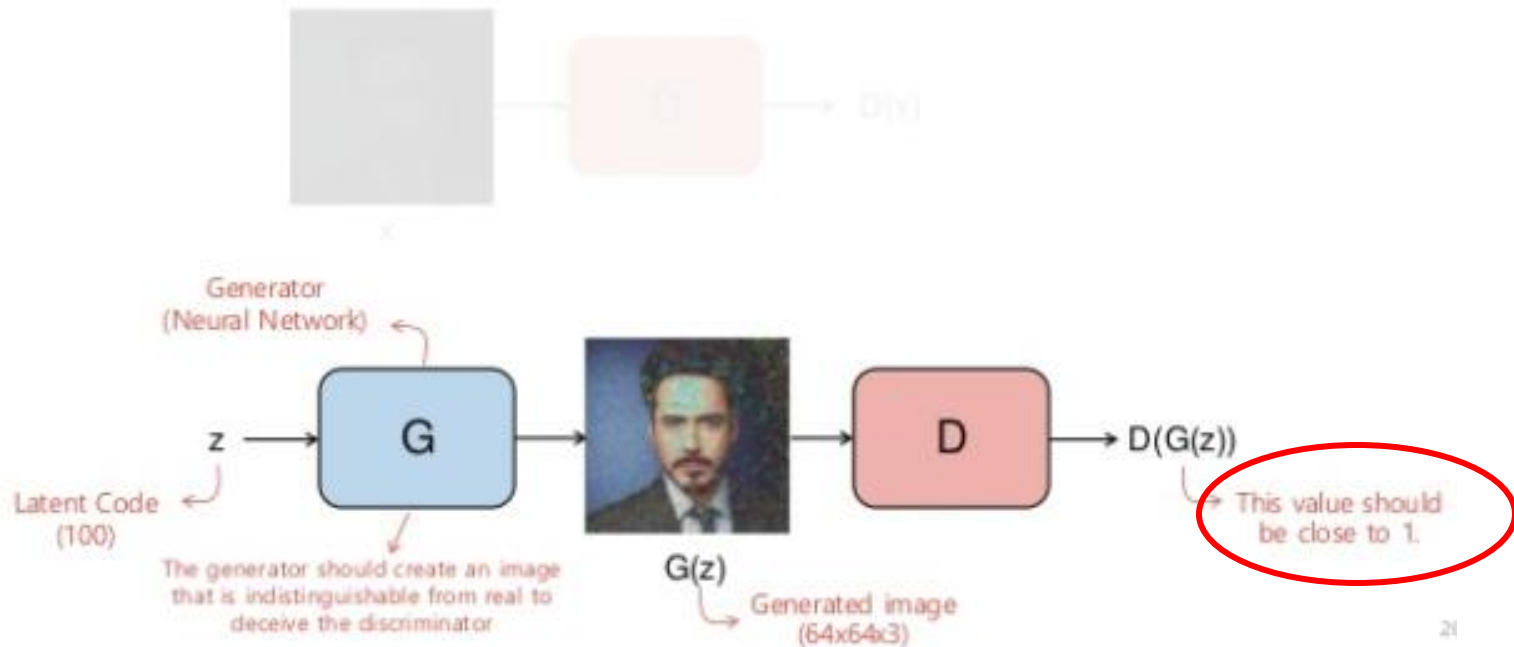
- D 의 입장



- G가 만든 가짜 이미지가 들어올 경우 0 을 내보내는 것

GAN 의 세부 내용

- G 의 입장



- G 자신이 만든 가짜 이미지를 D에 넣었을 때, 결과가 1이 나오도록 학습

수식으로 설명한 GAN

Sample x from real data distribution

Sample latent code z from Gaussian distribution

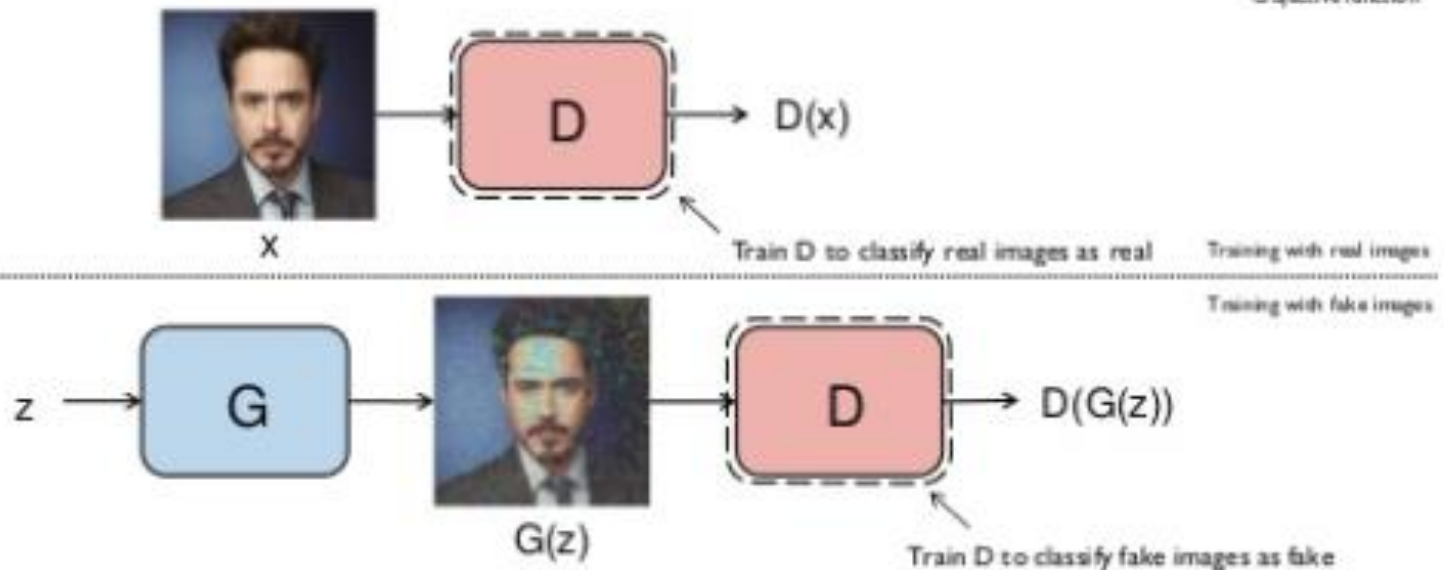
$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

D should maximize $V(D, G)$

Maximum when $D(x) = 1$

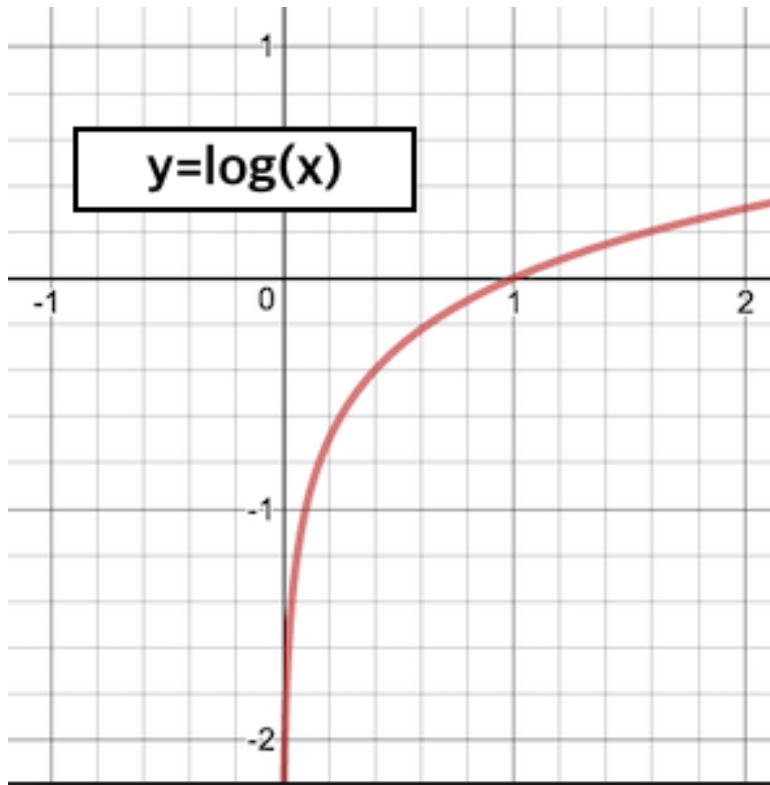
Maximum when $D(G(z)) = 0$

Objective function



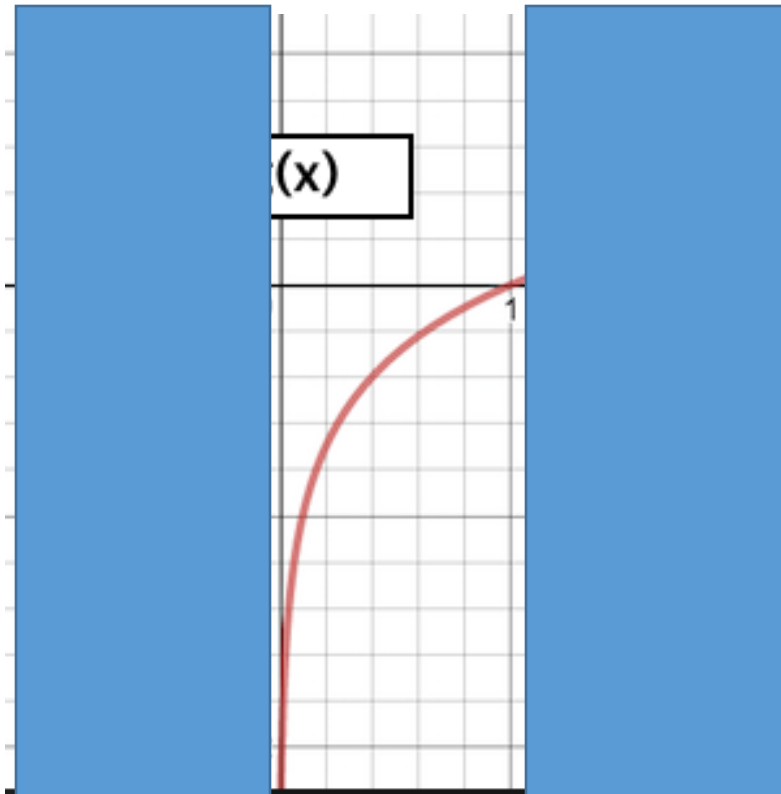
<https://www.slideshare.net/NaverEngineering/1-gangenerative-adversarial-network>

수식으로서 설명한 GAN



- G는 -1 ~ 1 사이의 값을 갖게 되고,
(normalization / 정규화)
- D는 앞서 설명한 것과 같이 0 과 1 두 가지 값을 갖게 되므로

수식으로로서 설명한 GAN



- G 는 $-1 \sim 1$ 사이의 값을 갖게 되고, (normalization / 정규화)
- D 는 앞서 설명한 것과 같이 0 과 1 두 가지 값만을 갖게 되므로
- 전체 $\text{Log}(x)$ 그래프에서 좌측 부분 내에서만 고려하면 됨
- $\text{Log}(x)$ 가 가장 큰 값을 갖기 위해서는 $x = 1$ 일 때,
- $\text{Log}(x)$ 가 가장 작은 값을 갖기 위해서는 $x = 0$ 일 때,

수식으로 설명한 GAN

G와 D가 있는데 뒤에 나오는 수식을
G는 최소화, D는 최대화하겠다는 의미

Sample x from real data distribution Sample latent code z from Gaussian distribution

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

D should maximize $V(D, G)$ Maximum when $D(x) = 1$ Maximum when $D(G(z)) = 0$

Objective function



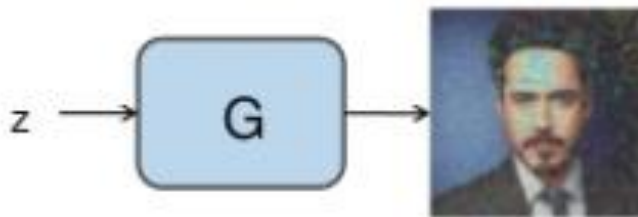
x



D(x)

Train D to classify real images as real

Training with real images



G(z)



D(G(z))

Train D to classify fake images as fake

Training with fake images

수식으로 설명한 GAN

D의 입장에서 다음 수식이 MAX가 되려면

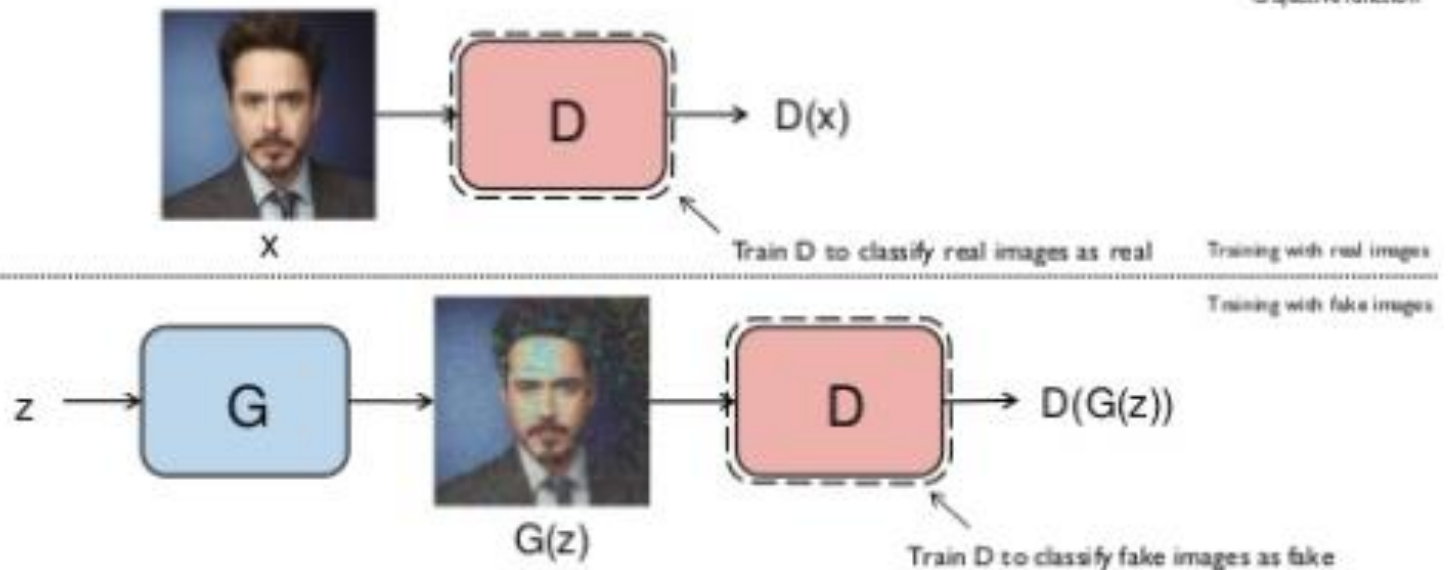
- 좌측 수식의 $D(x) = 1$
- 우측 수식의 $D(G(z)) = 0$

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

Annotations in the diagram:

- Sample x from real data distribution (points to x in $p_{data}(x)$)
- Sample latent code z from Gaussian distribution (points to z in $p_z(z)$)
- D should maximize $V(D, G)$ (points to \max_D)
- Maximum when $D(x) = 1$ (points to $\log D(x)$)
- Maximum when $D(G(z)) = 0$ (points to $\log(1 - D(G(z)))$)

Objective function



수식으로서 설명한 GAN

G의 입장에서 다음 수식이 MIN이 되려면

- 좌측 수식의 경우 고려하지 않음
- 우측 수식의 $D(G(z)) = 1$

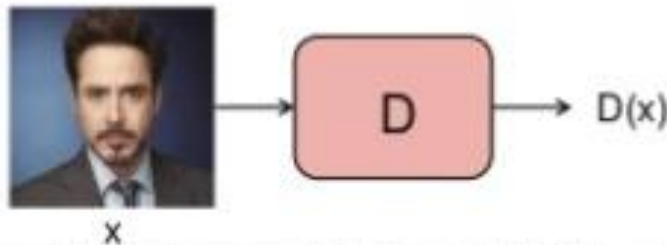
$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

G is independent of this part

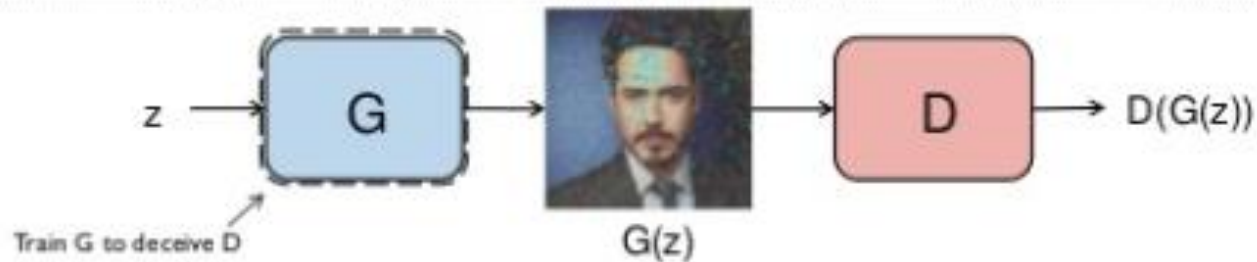
G should minimize $V(D, G)$

Minimum when $D(G(z)) = 1$

Objective function



Training with real images



Training with fake images

코드로서 설명한 GAN

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

Binary Cross Entropy Loss $(h(x), y)$

$$-y \log h(x) - (1 - y) \log(1 - h(x))$$

- Y 값이 0일 경우 $-\log(1-h(x))$
- Y 값이 1일 경우 $-\log h(x)$
- **D의 입장**에서 최종 loss 계산
- $-\log D(x) - \log(1 - D(G(z)))$
- 원래 수식에 마이너스가 붙어 있는 형태이므로 D가 최대 값을 가지려면 위 loss가 최소가 되면 됨

```
# Binary cross entropy loss and optimizer
criterion = nn.BCELoss()
d_optimizer = torch.optim.Adam(D.parameters(), lr=0.0003)
g_optimizer = torch.optim.Adam(G.parameters(), lr=0.0003)
```

```
outputs = D(images)
d_loss_real = criterion(outputs, real_labels)
```

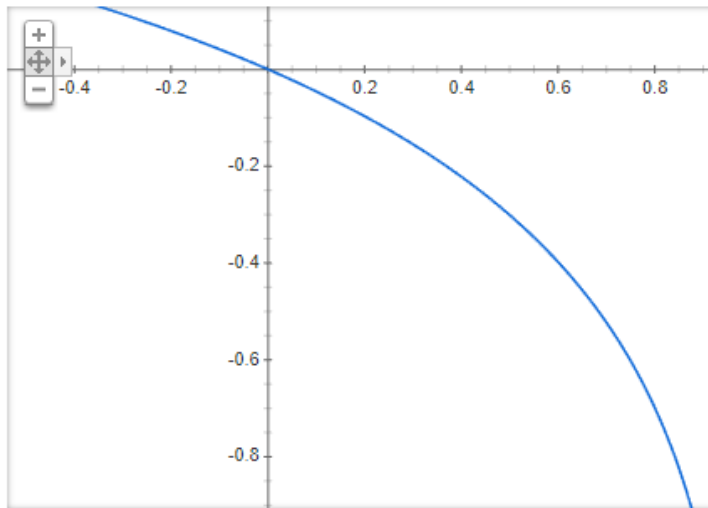
```
z = to_var(torch.randn(batch_size, 64))
fake_images = G(z)
outputs = D(fake_images)
d_loss_fake = criterion(outputs, fake_labels)
```

```
d_loss = d_loss_real + d_loss_fake
```

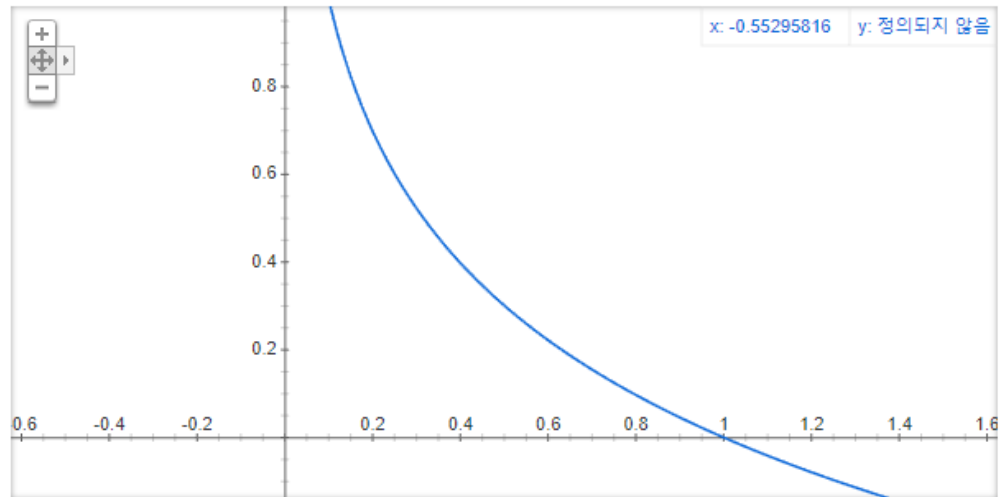
코드로서 설명한 GAN

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

log(1-x) 그래프



-log(x) 그래프



```
z = to_var(torch.randn(batch_size, 64))  
fake_images = G(z)  
outputs = D(fake_images)
```

- **G의 입장**에서 최종 loss 계산
- -log(D(G(z))) ← `g_loss = criterion(outputs, real_labels)`
- 원래 수식인 $\log(1-D(G(z)))$ 이 최소가 되는 것과 $-\log(D(G(z)))$ 가 최소가 되는 것은 동치 두 식 모두 $D(G(z))$ 가 1에 가까워야만 최소 값을 가짐

<https://www.slideshare.net/NaverEngineering/1-gangenerative-adversarial-network>

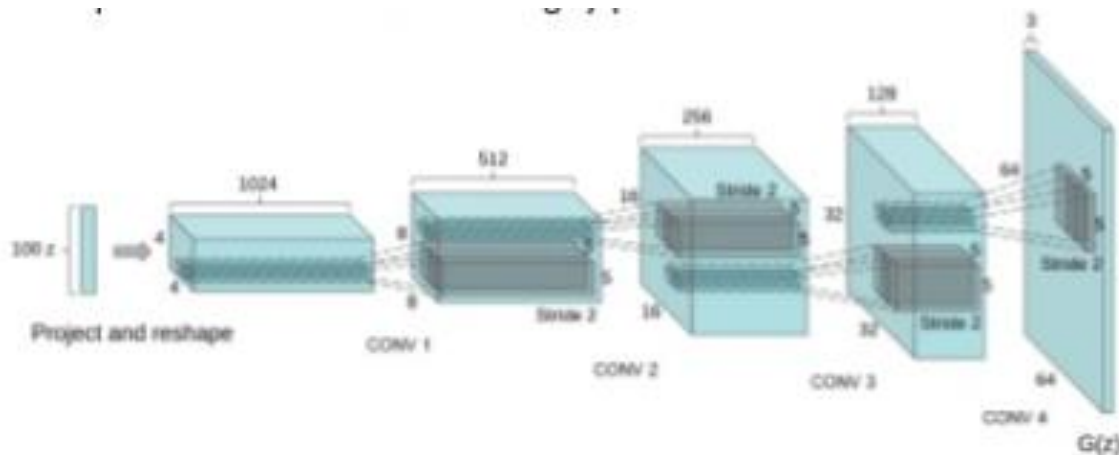
DCGAN의 개념

■ GAN의 한계점

- 학습이 쉽지 않음
- 초반에 G가 성능이 너무 좋아지면, D의 성능이 확 떨어지고, (D가 멍청하니 잘된건지 알 길이..)
- 반대로 D의 성능이 너무 좋아지면, G의 성능이 확 떨어지는 등의 문제 존재

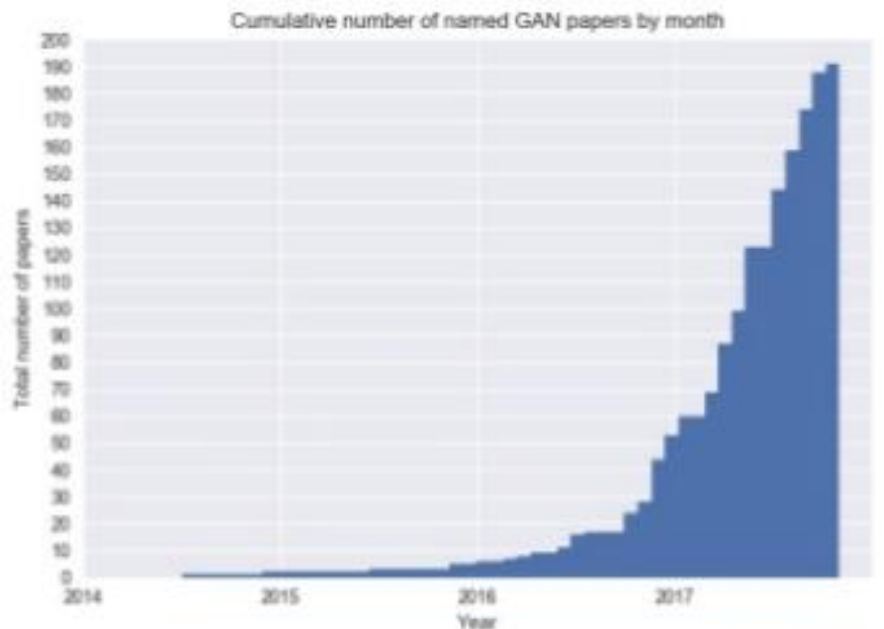
■ DC (Deep Convolutional) GAN

- GAN의 학습이 어려워 힘든 문제가 있는 반면
- DCGAN의 경우 간단한 네트워크로 안정적으로 쉽게 GAN 구현 가능



DCGAN 의 개념

▪ DCGAN 이후 상황



Explosive growth—All the named GAN variants cumulatively since 2014. Credit: [Boris Ginsburg](#)

- GAN 이 2014년 말에 나왔으나 학습이 쉽지 않아 외면 받았음
- 그러다 2015년 DCGAN 이 나온 이후 GAN 관련 논문이 폭발적으로 증가
- DCGAN 의 골자는 “Conv net 을 이용하되 파라미터를 자신들이 언급한 대로 써라” 임

(정말 하라는 대로 하면 잘됨)

<https://www.slideshare.net/NaverEngineering/ai-83896428>

PixToPix 의 개념

■ 핵심 알고리즘

$$\sum_{(x,y)} \|y - G(x)\|_1 + L_{GAN}(G(x), y)$$

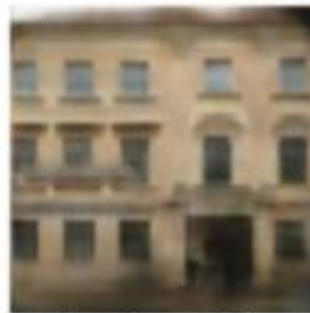
“이런 사진 주면 저렇게 만들어줘” 와 같은 개념



Input



Ground Truth



L1 loss only

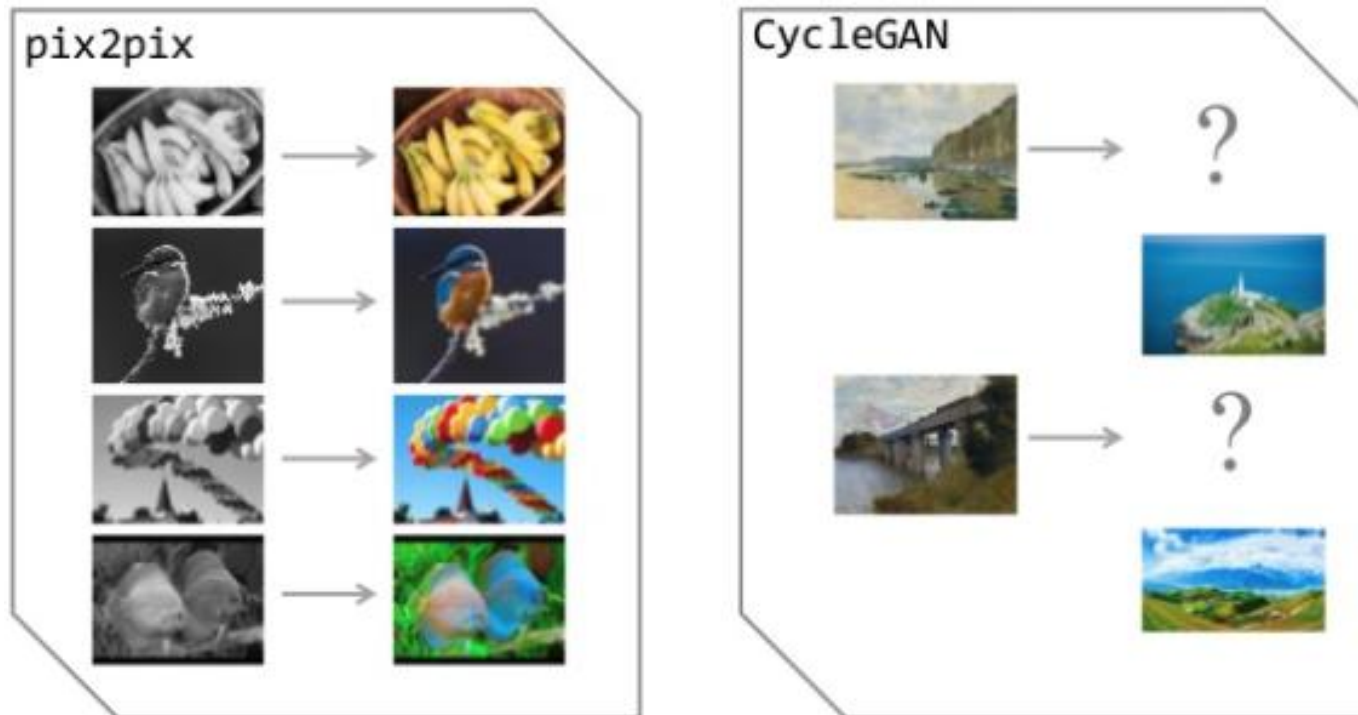


L1+GAN loss

- 입력 값과 **Pair** 한 형태의 **GT** 값을 이용
- 초기 아이디어는 L1 loss 만을 이용하여 흑백을 컬러 이미지로 바꾸려는 형태
- 하지만 전반적으로 흐릿한 이미지가 나와 **GAN** 을 통해 조금더 **G** 가 나온 이미지를 만들도록 학습

Cycle GAN 의 개념

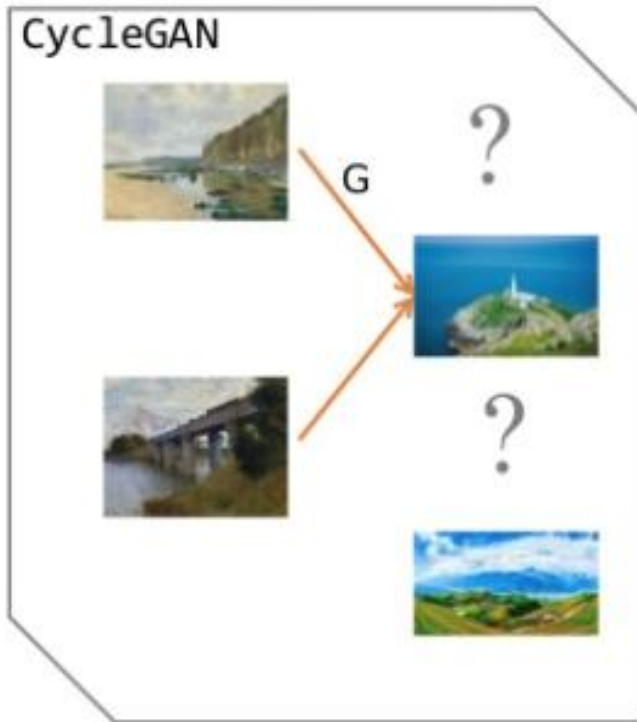
- 핵심 아이디어



- pair 한 이미지 데이터 셋으로는 적용에 한계점 존재
- unpair 한 이미지 데이터 셋을 활용한 방안을 없을까? 에서 시작

Cycle GAN 의 개념

▪ Cycle loss 의 필요성



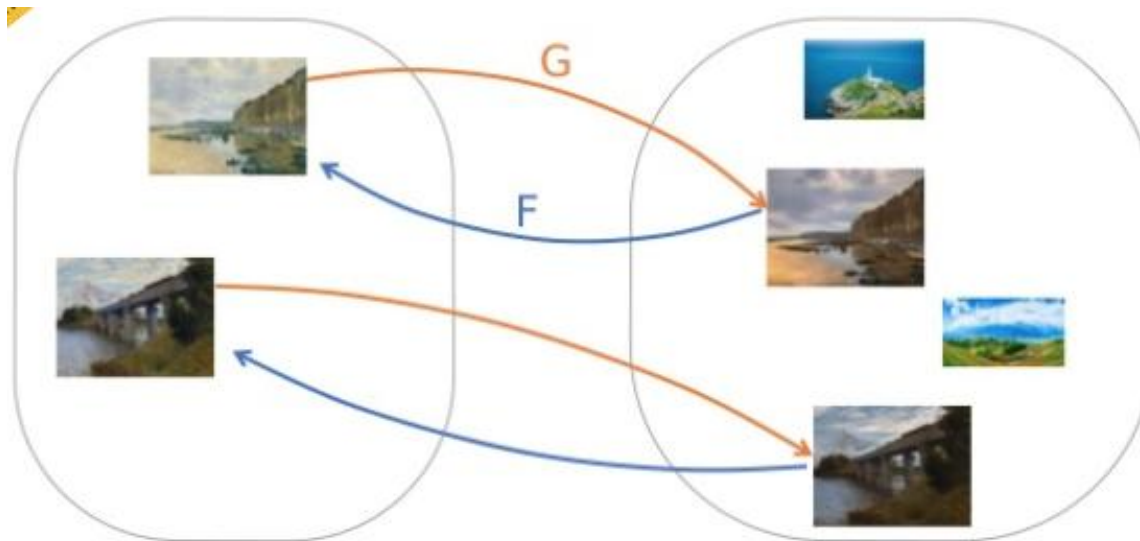
우리가 하고 싶은 것은 원본 형태는 유지한 채 속성만 바꾸고 싶음

- 단순히 “그림 → 사진” 혹은 “야간 → 주간”으로의 변경에 대한 loss 만 존재한다면 좌측 그림과 같이 어떠한 입력을 넣어도 특정 이미지 1장만 출력하는 **mode collapse** 발생
- 그렇기 때문에 원래 이미지를 변경하되 다시 되돌렸을 때, 원본이미지가 나오도록 하는 **Cycle loss** 를 적용해 **mode collapse** 를 해결

<https://www.slideshare.net/NaverEngineering/ai-83896428>

Cycle GAN 의 개념

■ 핵심 알고리즘

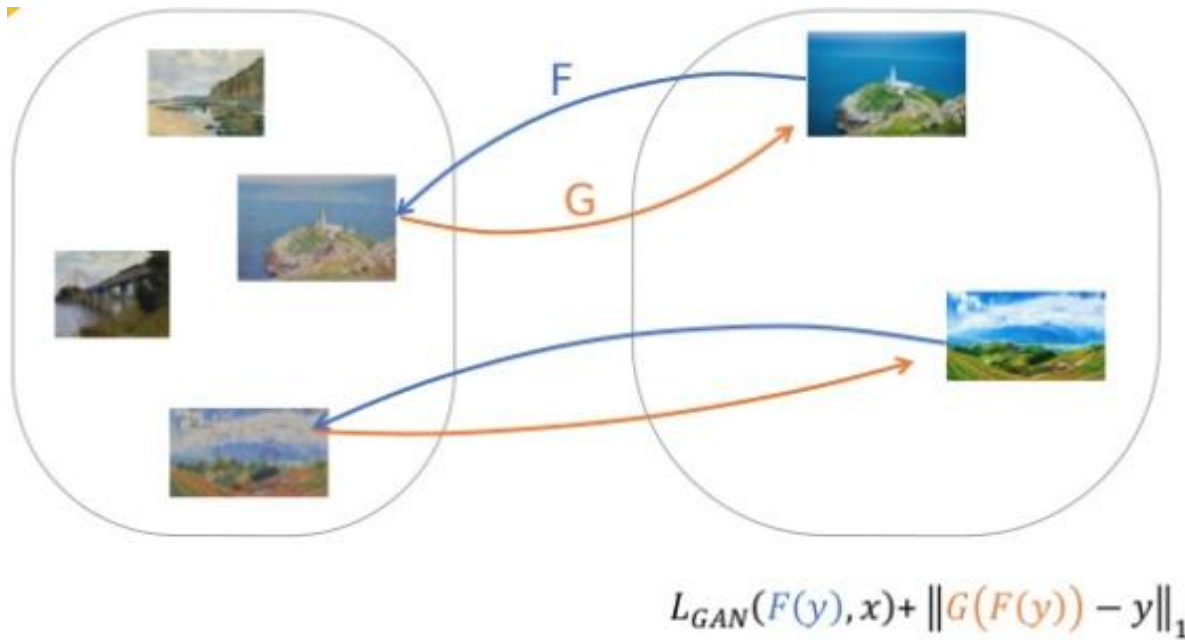


$$L_{GAN}(G(x), y) + \|F(G(x)) - x\|_1$$

- A 타입 사진을 B 타입 사진으로 바꿔줘 (좌측 식)
- 하지만 다시 원복을 했을 때, 원래 이미지가 나와야 해 (우측 식)

Cycle GAN 의 개념

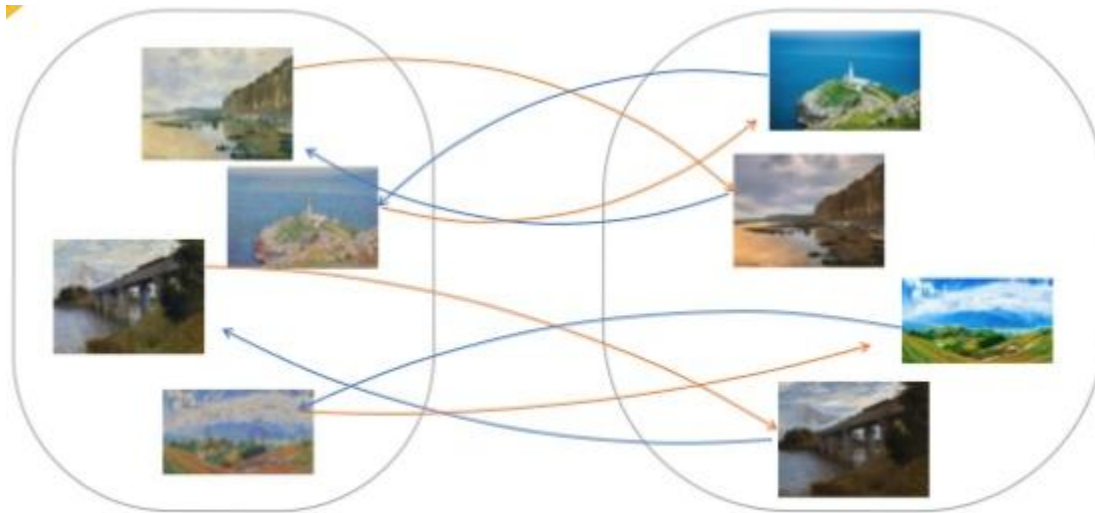
■ 핵심 알고리즘



- 반대로 B 타입 사진을 A 타입 사진으로 바꿔줘 (좌측 식)
- 하지만 다시 원복을 했을 때, 원래 이미지가 나와야 해 (우측 식)

Cycle GAN 의 개념

- 핵심 알고리즘

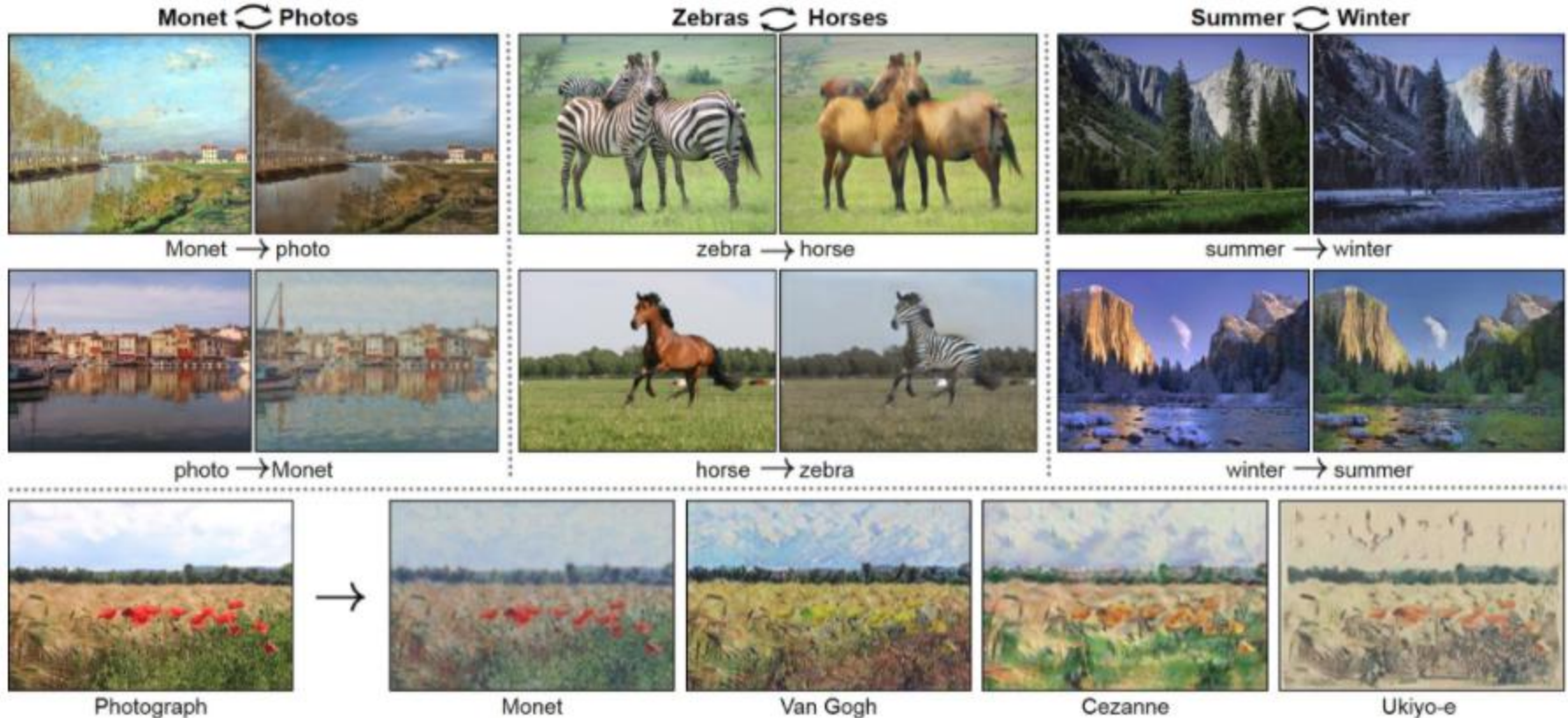


$$L_{GAN}(G(x), y) + \|F(G(x)) - x\|_1 + L_{GAN}(F(y), x) + \|G(F(y)) - y\|_1$$

- 양방향에 처리에 대한 loss 를 더해주면 최종 수식 완성

Cycle GAN 의 개념

- 이미지 간의 속성 전환



- 다양한 형태의 속성 전환 가능

테스트 결과물 (2)

■ 야간 원본 사진



real_B-0



real_B-1



real_B-2



real_B-3



■ 주간으로 속성을 변경한 사진



fake_samples-final_BtoA-0



fake_samples-final_BtoA-1



fake_samples-final_BtoA-2



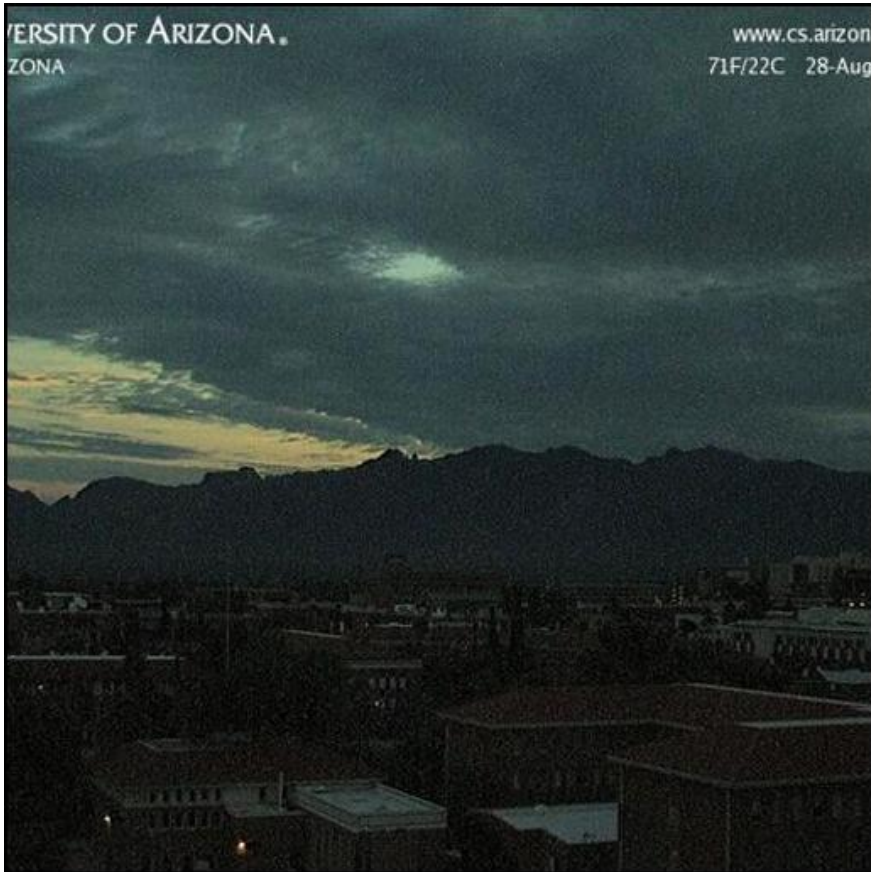
fake_samples-final_BtoA-3



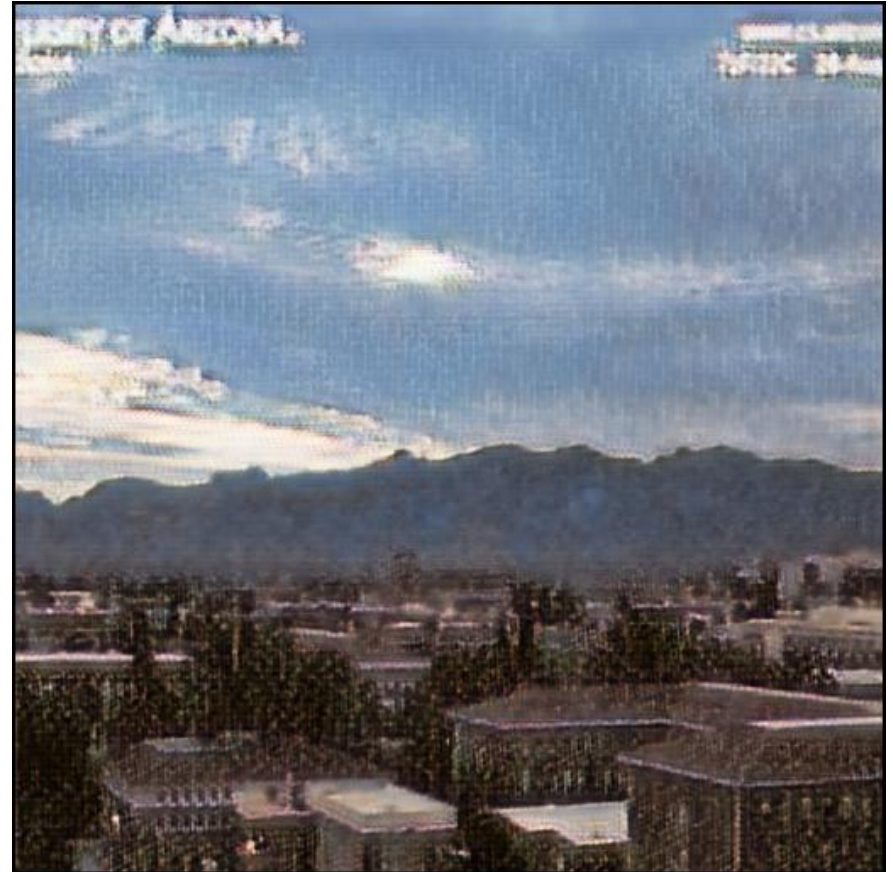
- Cycle GAN (Generative Adversarial Network) 를 활용한 결과
- 작은 그림으로 얼핏 보면 속성이 잘 변환 된 듯 함

테스트 결과물 (2)

- 야간 원본 사진



- 타입을 변경한 사진



- 하지만 큰 사진으로 보면 이미지가 원본에 비해 뭉게진다는 것을 알 수 있음

테스트 결과물 (2)

- 야간 원본 사진



- 타입을 변경한 사진



- 뭉게지는 것 이외에 **없던 차 정보**가 생기는 등의 문제 발생

Cycle GAN 의 특징 정리

- 데이터의 개수 보다는 **다양한 형태의 데이터 필요**



- 좌측의 경우 낮에는 항상 앞에 차가 있는 데이터가 많이 때문에 발생
 - 우측의 경우에는 말을 학습 시킬 때 사람이 탑승 하고 있는 사진이 1장 밖에 없었음
 - 만일 야간 영상을 주간으로 변경하는 학습을 하는데 **야간 학습 영상에 보행자가 없었고, 입력 영상에는 보행자가 있을 경우 변경된 주간 영상 내에는 보행자가 지워질 확률이 매우 높음**
- **Pair** 하지 않은 데이터로 학습 하기 위해 만들어졌지만 **데이터가 Pair 할 수록 좋음**
 - 오래 학습 시킨다고 해서 좋은 성능을 보장하지 않음
 - **Identity loss** 의 경우 **unpair** 데이터 셋에 대한 보정을 해줌

Cycle GAN 의 특징 정리

- Identity loss 의 경우 unpair 데이터 셋에 대한 보정을 해줌



n02381460_1300_fake_B



n02381460_1300_real_A



n02381460_1350_fake_B



n02381460_1350_real_A



n02381460_1360_fake_B



n02381460_1360_real_A



n02381460_1420_fake_B



n02381460_1420_real_A



n02381460_1540_fake_B



n02381460_1540_real_A

원논문 저자의 말
데이터 셋 결과



우리가 구현한 초기
구현 결과물



Identity loss 추가 후
결과물

결론

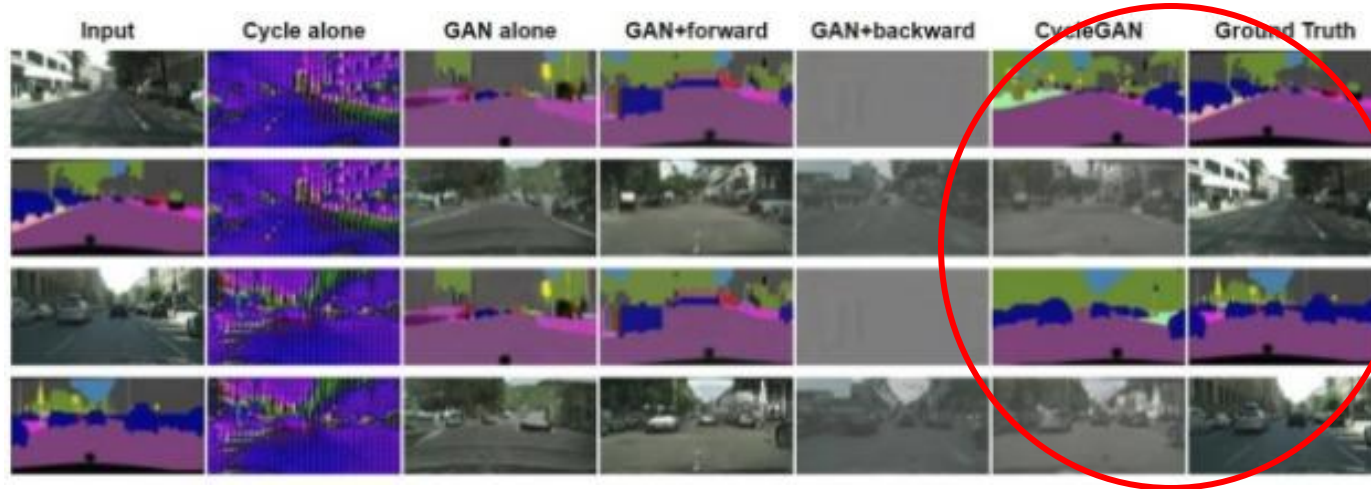
- 최신 ICT 이슈 내에서 발췌

구글의 인공지능(AI) 연구부문인 구글 브레인의 이언 굿펠로우가 2014년 발표한 GAN(Generative Adversarial Networks)은 최근 10년간 AI 연구에서 가장 뛰어난 성과로 평가 받고 있으며, 현재 약 200여 개의 변형된 기술들이 연구되고 있음. GAN은 딥러닝과 달리 인간의 개입 없이도 그리고 학습할 데이터가 없어도 신경망 스스로 학습해 나갈 수 있음을 보여주고 있다는 점에서 큰 기대를 받고 있으며, 그 잠재력만큼 동시에 큰 위험도 내포하고 있어 우려의 목소리도 높아지고 있음

- AI 계에 혁신을 불러다 준 기술은 맞으나 당장 우리의 솔루션에 적용은 어려울 듯 함
- 그리고 앞서 언급한 없는 객체를 만든다던 지 사라지게 한다던 지의 이슈는 안전을 우선으로 하는 ADAS 에서 사용하기에는 치명적인 약점

결론

- 다른 방향으로의 활용 (원 논문 저자가 언급한 부분이기도 함)



- Object detection 에 이어 semantic segmentation 이 각광 받고 있는데 이와 관련한 GT 데이터 생성 등에 활용이 가능할 듯 함