윤세이 마코토

16기 김성윤 박준민 양지은 유승수 장현우 조석주

프로젝트 목표

일반 사진

Image Style Transfer

애니메이션 스타일

프로젝트 목표

일반 사진

HOW?

Image Style Transfer

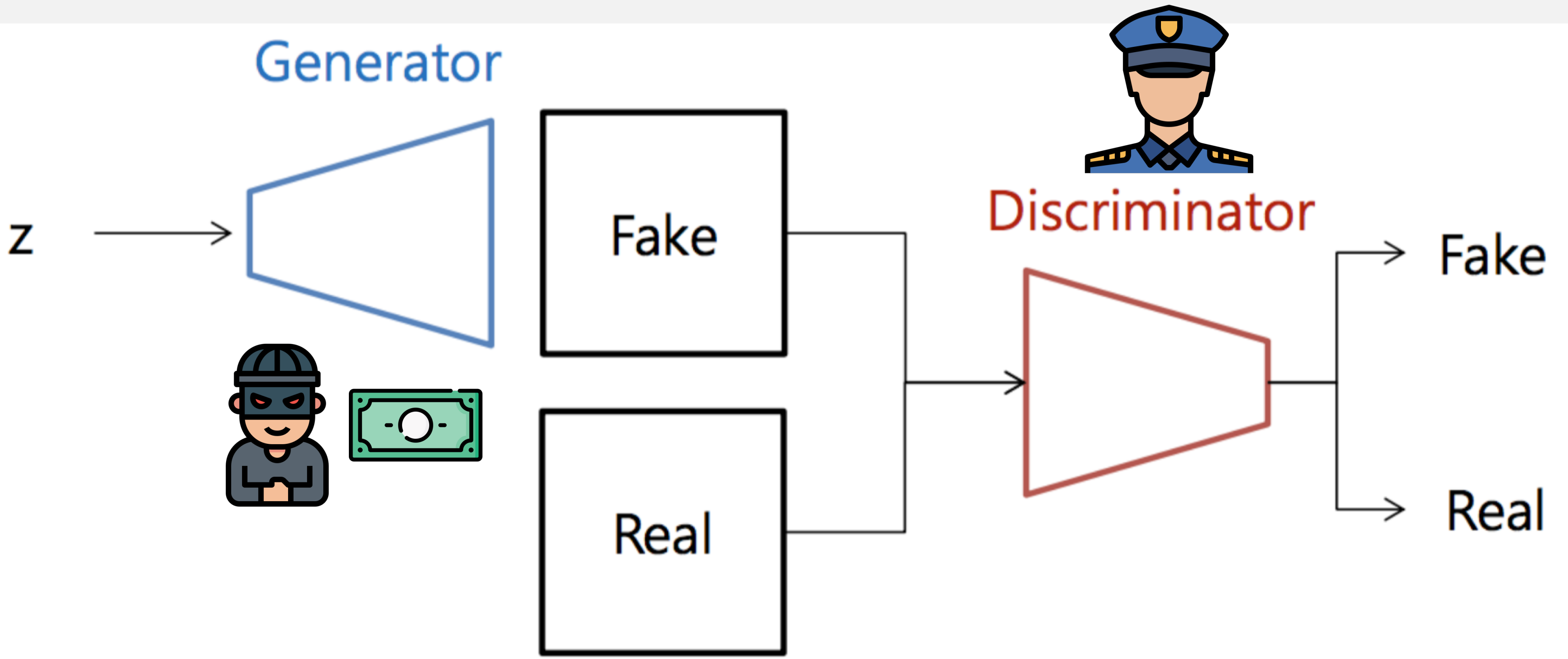
애니메이션 스타일

Basic Concepts

1. Vanilla GAN
2. Conditional GAN
3. Cycle GAN

Vanilla GAN

(Generative Adversarial Network)



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

$$where\ p_z(z)\ denotes\ Zero - Mean\ Gaussian\ noise\ variable$$

Adversarial Loss의 의미

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

D (Discriminator)의 입장

실제 데이터(x)를 입력하면 $D(x)$ 가 높아져야 함.
반면, 가짜 데이터($G(z)$) 를 입력하면 $D(G(z))$ 가 낮아져야함.
즉, Loss를 크게 하는 방향으로 학습된다.

$\log D(x) \uparrow$

$\log(1 - D(G(z))) \uparrow$

G (Generator)의 입장

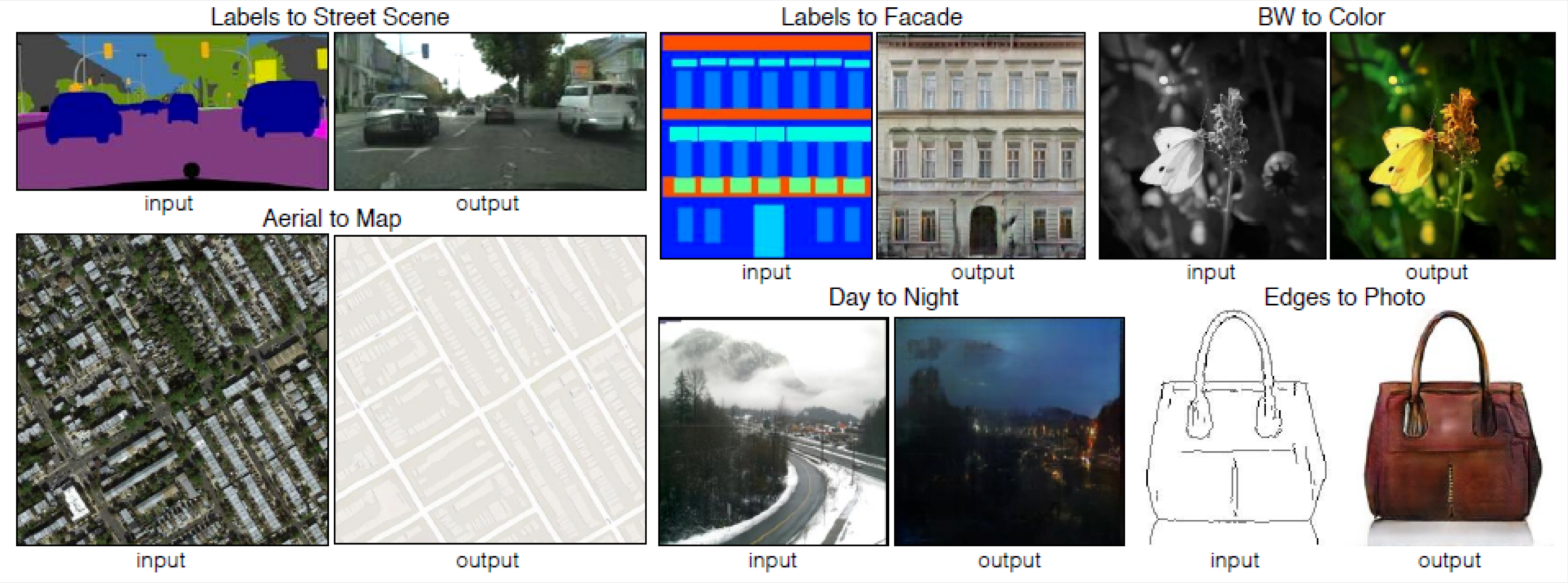
D는 고정된 함수이다.
Zero-Mean Gaussian으로 부터 생성된 stochastic한 가짜 데이터 $G(z)$를 D에 넣을때, D를 속여야 한다. 즉, $D(G(z))$를 높여야 한다.

$\log(1 - D(G(z))) \downarrow$

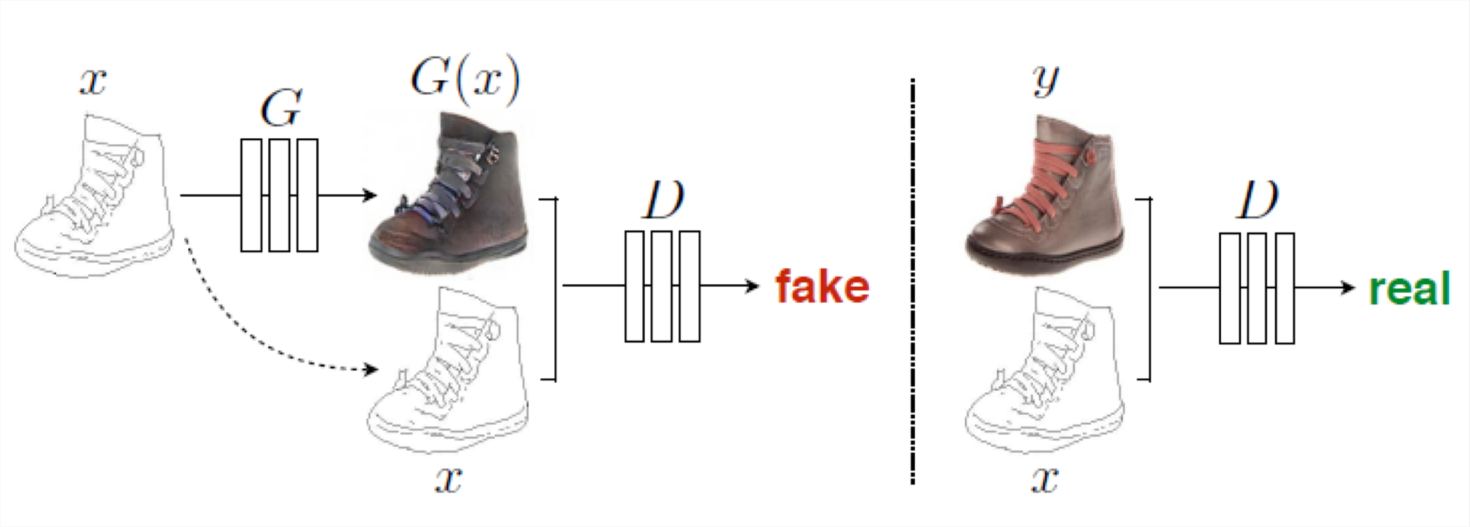
Generator가 만들어 내는 그림에 조건을 걸 수 없을까?

e.g. 연세대학교 사진을 바탕으로 목표하는 사진 생성하기

Conditional GAN (cGAN)



Conditional GAN (cGAN)



- Discrimitator는 $tuple\ \{synthesized\ by\ the\ generator, edge\}$와 $\{real\ photo, edge\}$를 구분할 수 있게 학습
- Generator는 x 와 $random\ noize\ vector\ z$를 입력으로 받아 fake image를 생성

Objective Function of cGAN

Adversarial Loss

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))]$$

L1 Loss

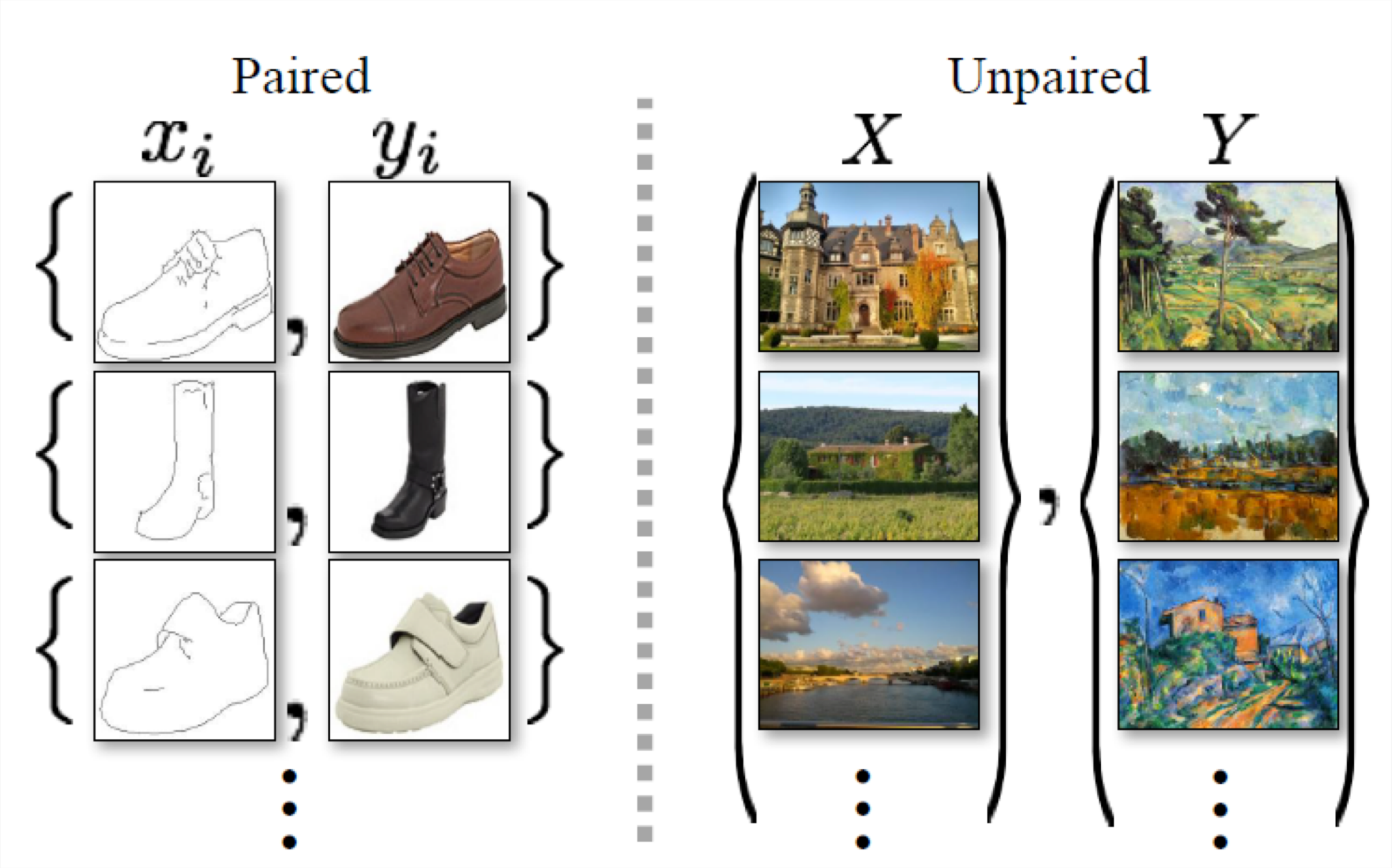
$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1]$$

Our final objective is:

$$G^*, D^* = arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

학습을 위해선 data의 pair $\{x_i, y_i\}$ 가 필요함에 유의!

Paired VS. Unpaired



Cycle GAN

사진

모네의 그림

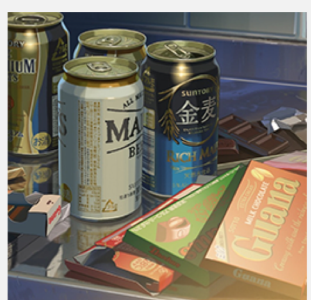
Unpaired 학습의 문제점

source domain X

$D(G(x))$

다 신카이마코토 맞아!!

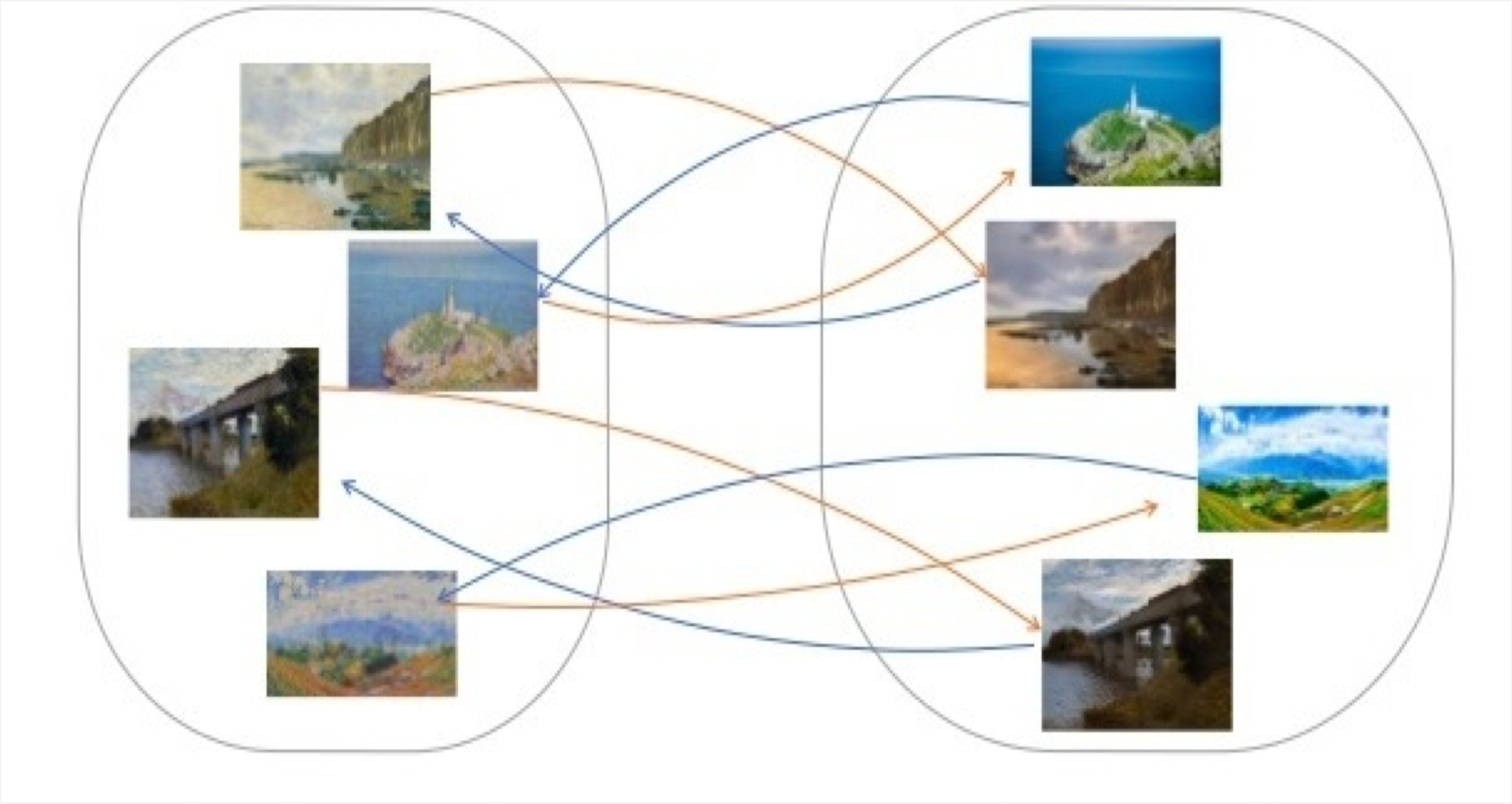
$G(x)\ x \epsilon X$

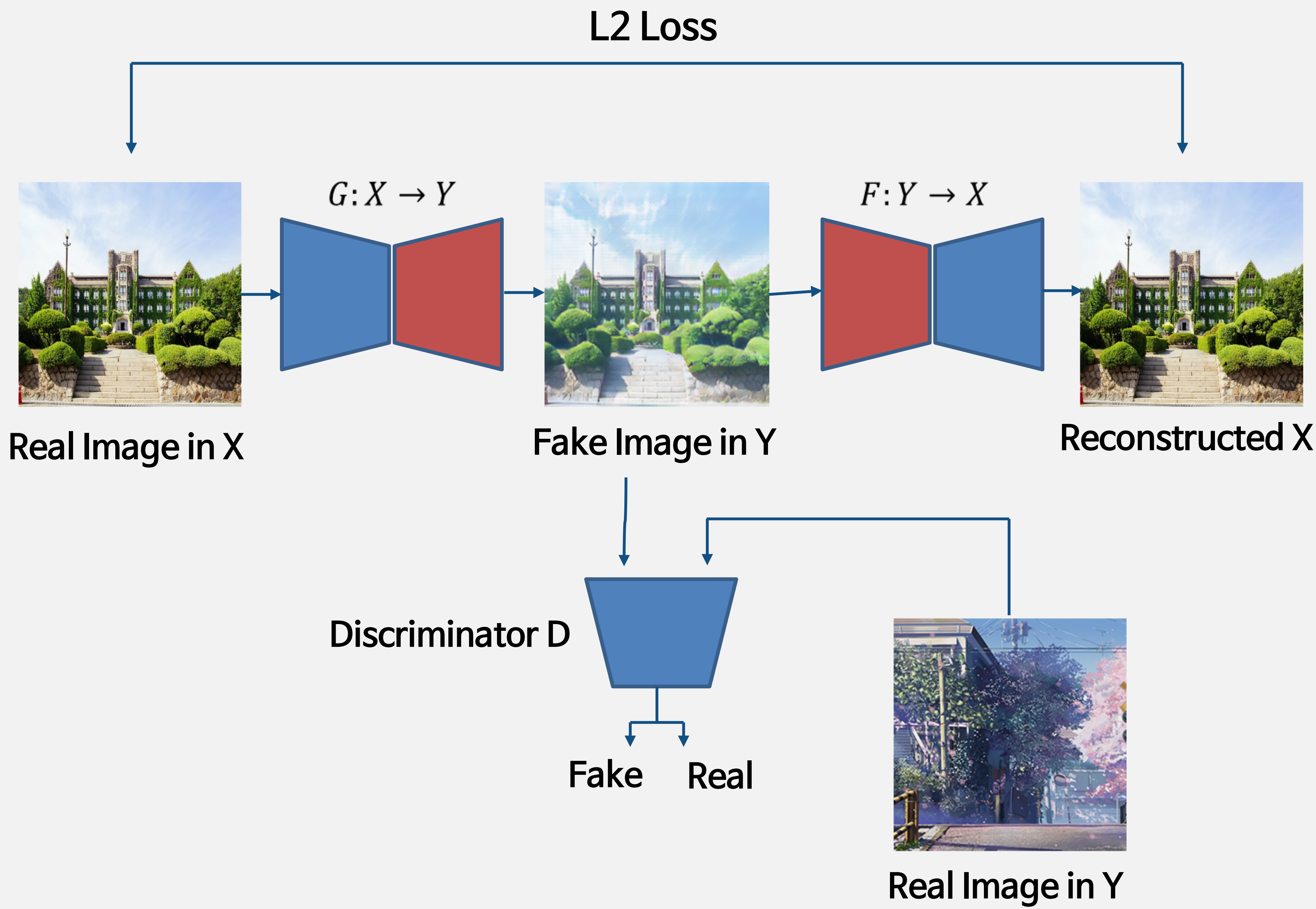


$G(x)\ x \epsilon X$ 의 distribution이 Y와 일치한다고 하더라도
$G(x)$ 가 우리가 원하는 바람직한 결과가 아닐 수 있음.

Cycle Consistency

X **Y**

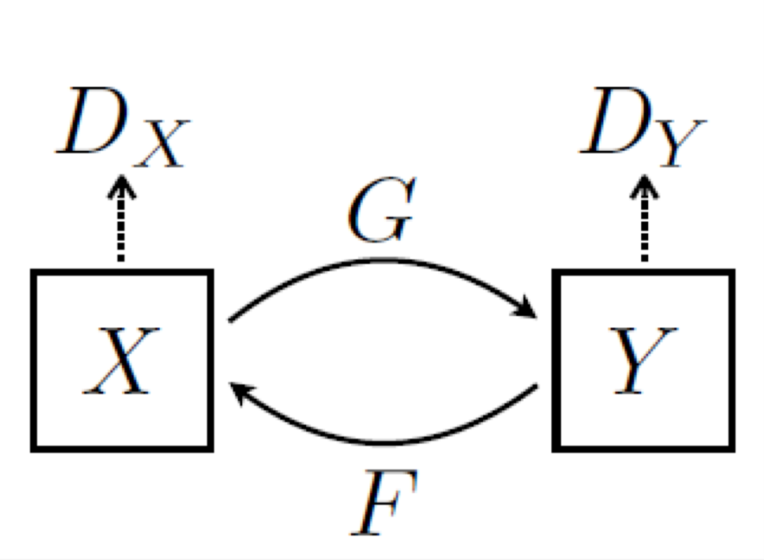



L2 Loss
$G: X \rightarrow Y$
$F: Y \rightarrow X$
Real Image in X
Fake Image in Y
Reconstructed X
Discriminator D
Fake
Real
Real Image in Y

$$G, F: Generator$$

$$D_X: Distinguishes\ x\ from\ F(y)$$

$$D_Y: Distinguishes\ y\ from\ G(x)$$



$given\ two\ domains\ X\ and\ Y, define\ G: X \rightarrow Y\ and\ F: Y \rightarrow X$

$Let\ D_X\ and\ D_Y\ as\ adversarial\ discrimin$
$where\ D_X\ aims\ to\ discriminate\ beyween$
$D_Y\ aims\ to\ discriminate\ between\ \{y\}\ an$

$Adversarial\ Loss: G, D_Y\ \ F, D_X$를 적대적으로 학습하기 위한 Loss.

$$\mathcal{L}_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x))]$$

$$\mathcal{L}_{GAN}(F, D_X, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{data}(x)}[\log(1 - D_Y(F(x))]$$

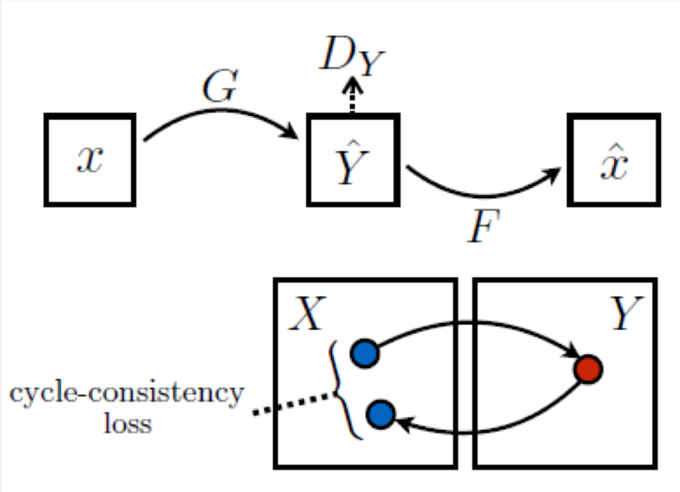
Cycle Consistency Loss: *Cycle consistency*를 강제하기 위한 Loss.

$$\mathcal{L}_{cyc}(G,F) = \mathbb{E}_{x\sim p_{data}(x)}\left[\left\|F(G(x)) - x\right\|_1\right]$$

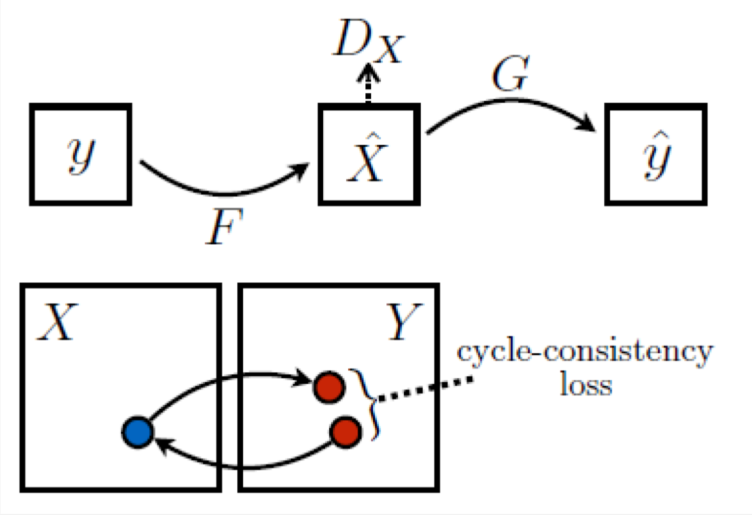
$$+\mathbb{E}_{y\sim p_{data}(y)}[\left\|G(F(y)) - y\right\|_1]$$

$$x \rightarrow G(x) \rightarrow F(G(x)) \cong x, y \rightarrow F(y) \rightarrow G(F(y)) \cong y$$

Forward cycle-consistency loss



Backward cycle-consistency loss

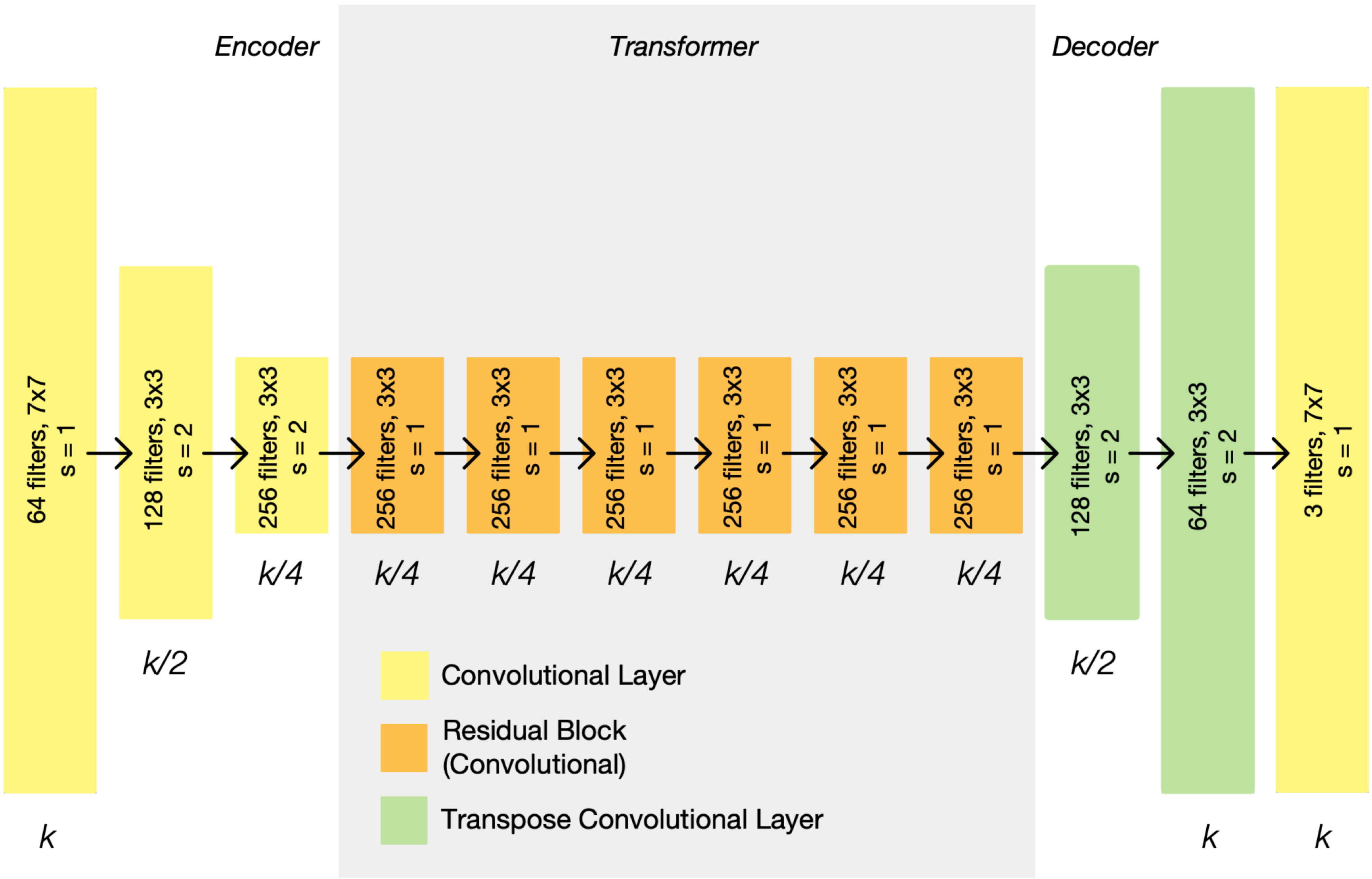


$Final\ Objective$

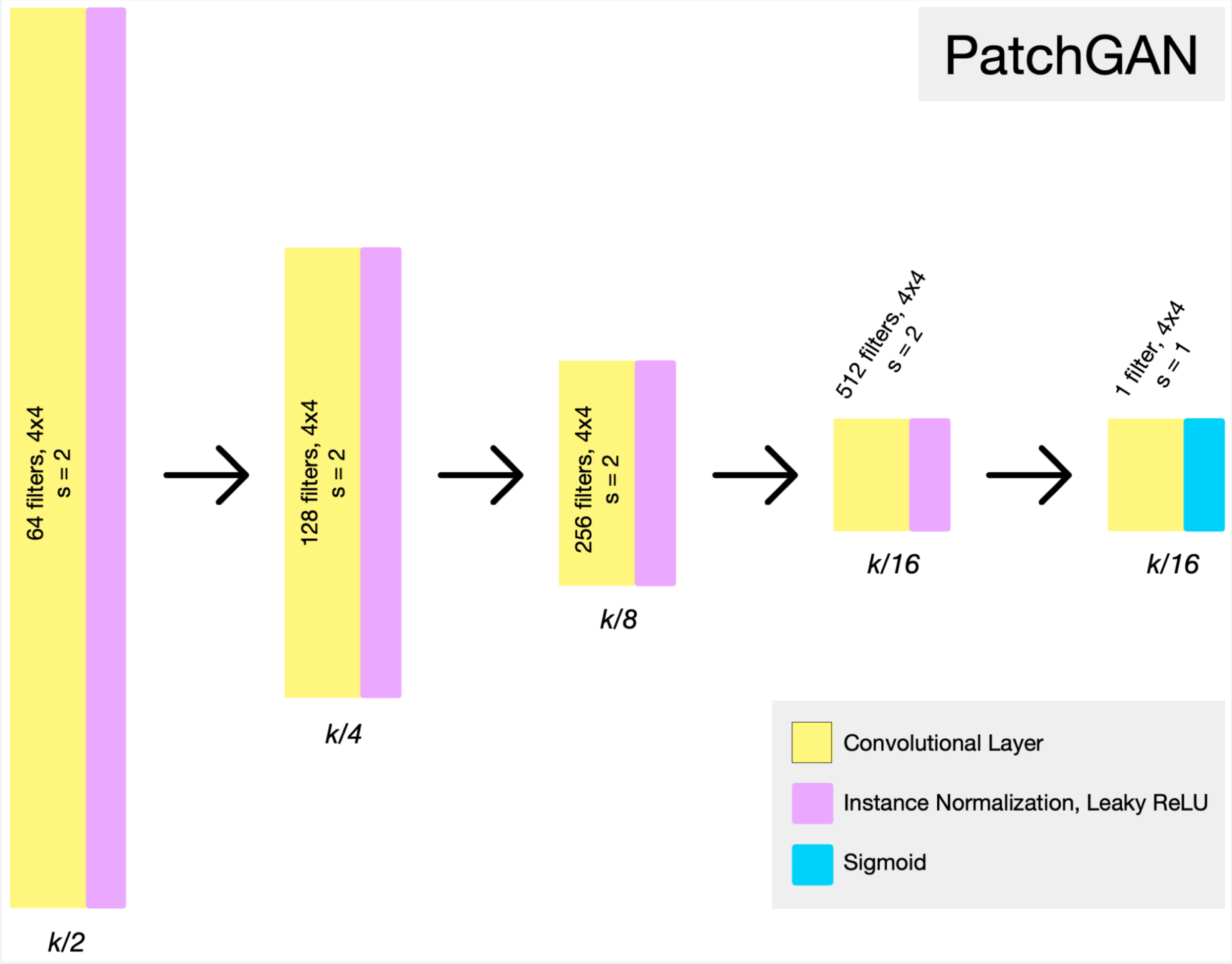
$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, X, Y) + \lambda\mathcal{L}_{cyc}(G, F)$$

$$G^*, F^* = arg \min_{G,F} \max_{D_X,D_y} \mathcal{L}(G, F, D_X, D_Y)$$

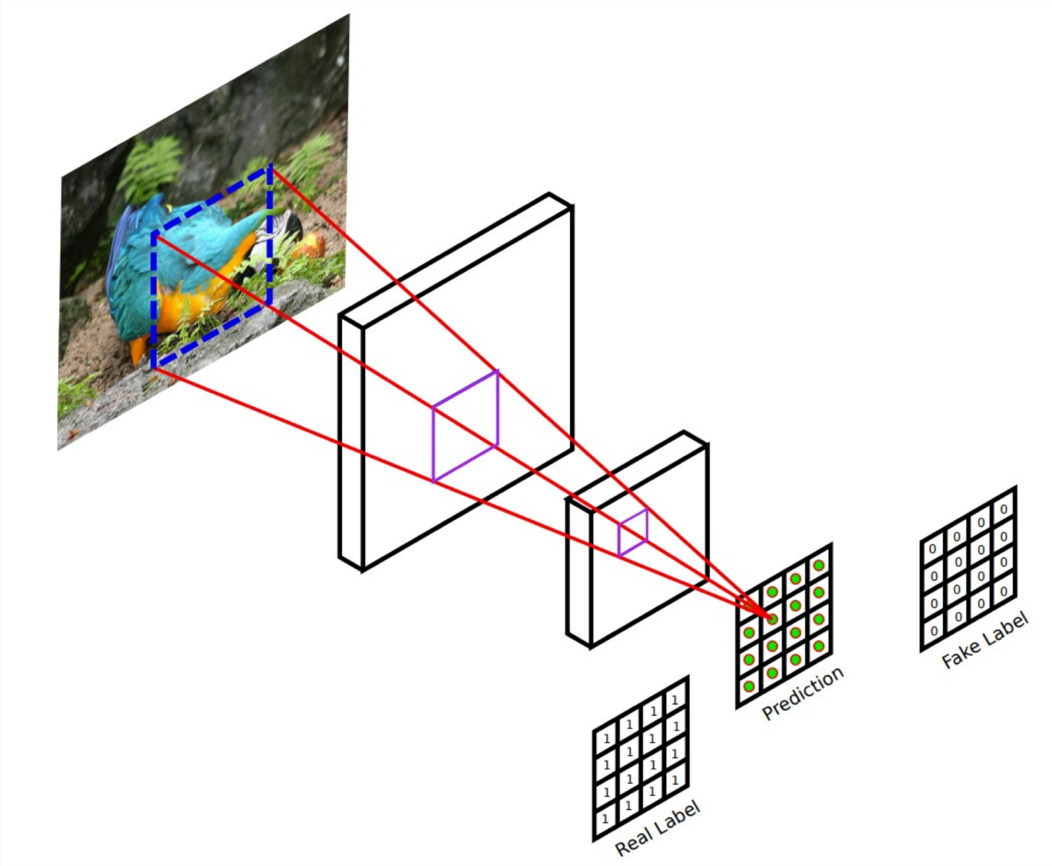
Generator Architecture



Discriminator Architecture



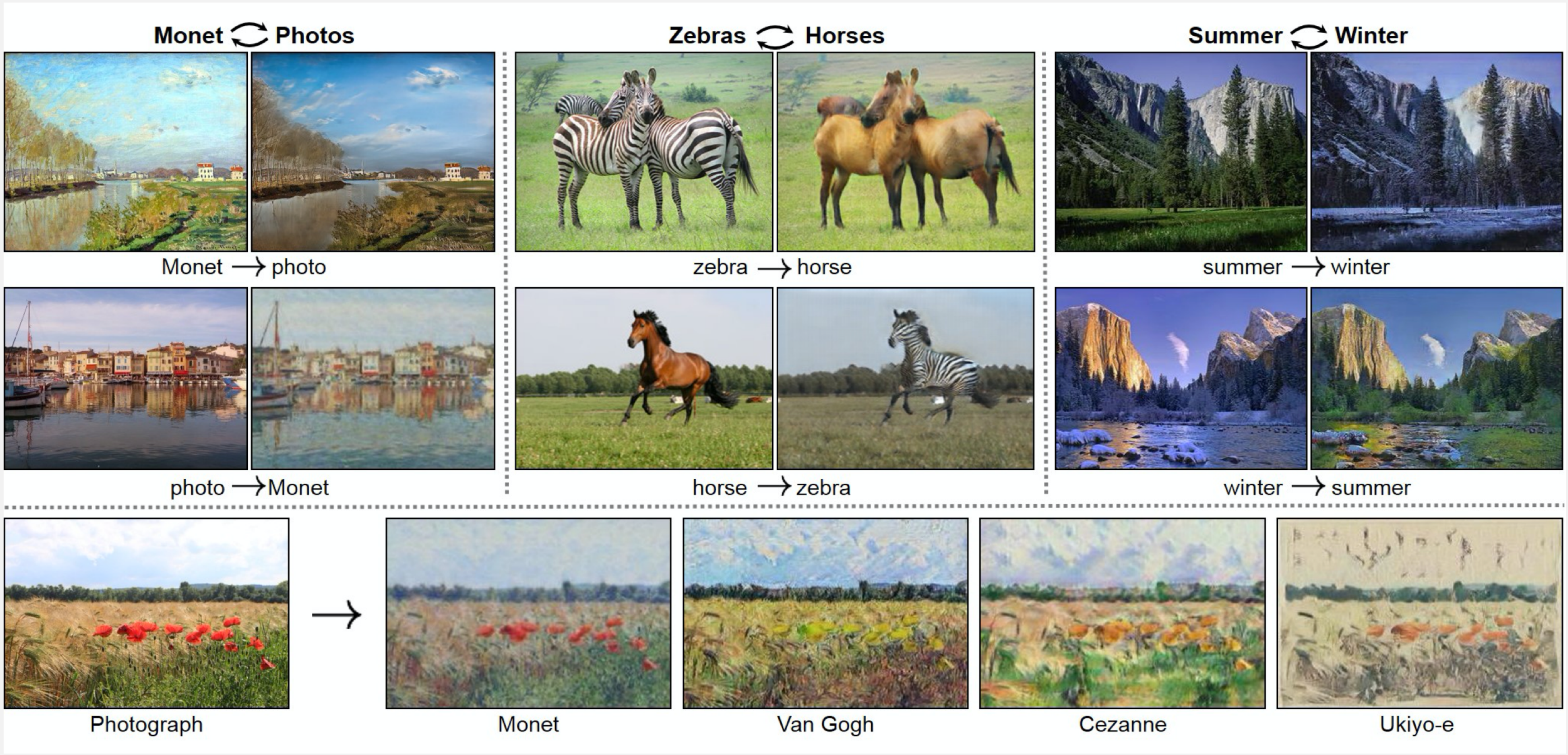
PatchGAN



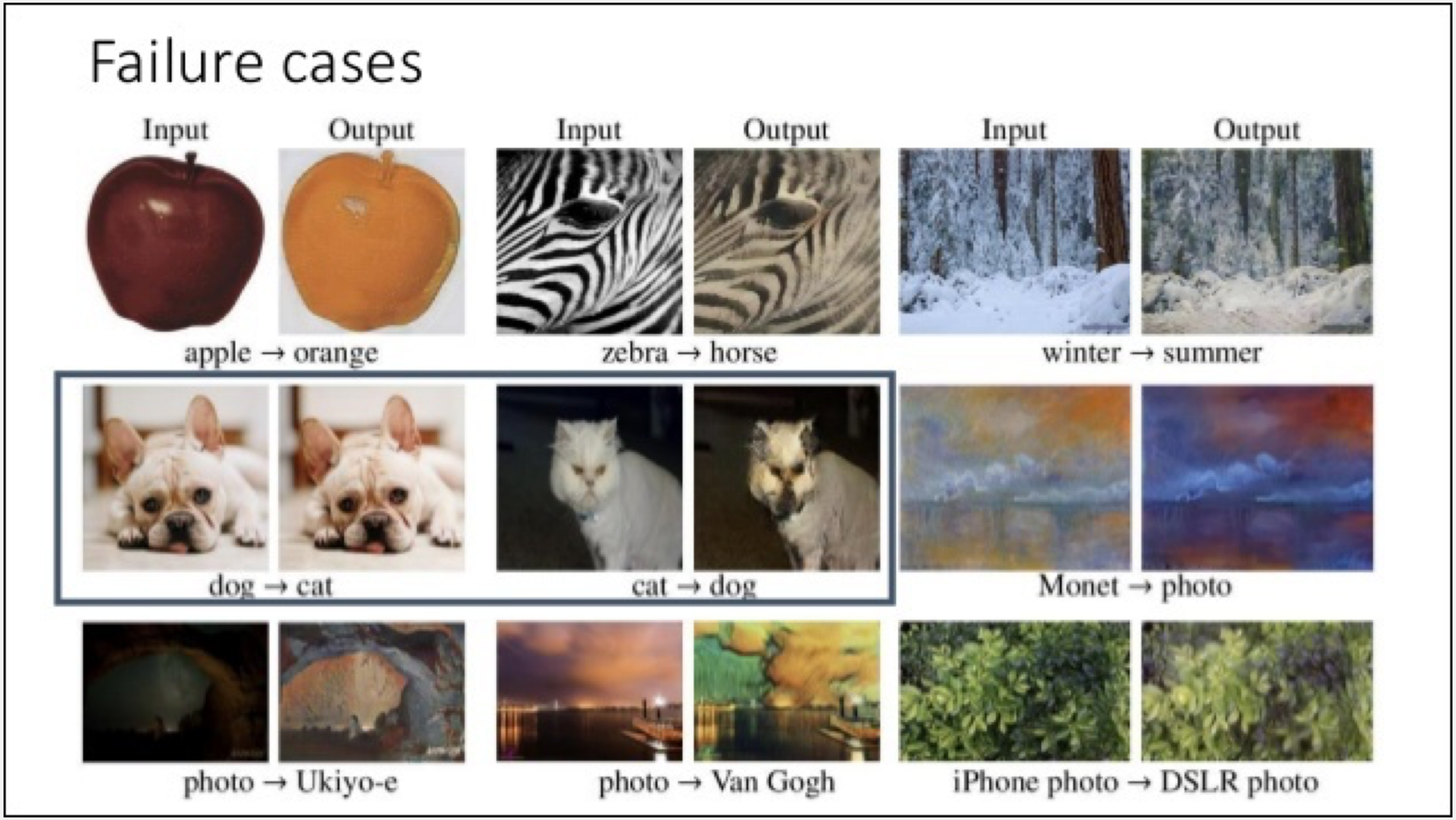
- 전체 영역이 아닌 특정 크기의 patch단위로 input의 진위여부를 판단

- Generator가 만들어준 입력이미지 전부를 보고 Real/Fake 여부를 판단하게 되면 D를 잘 속일 수 있는 방향으로 학습하게 되고, 이미지의 블러가 심해져

Cycle GAN으로 할 수 있는 것



Cycle GAN의 한계



Cycle GAN의 한계

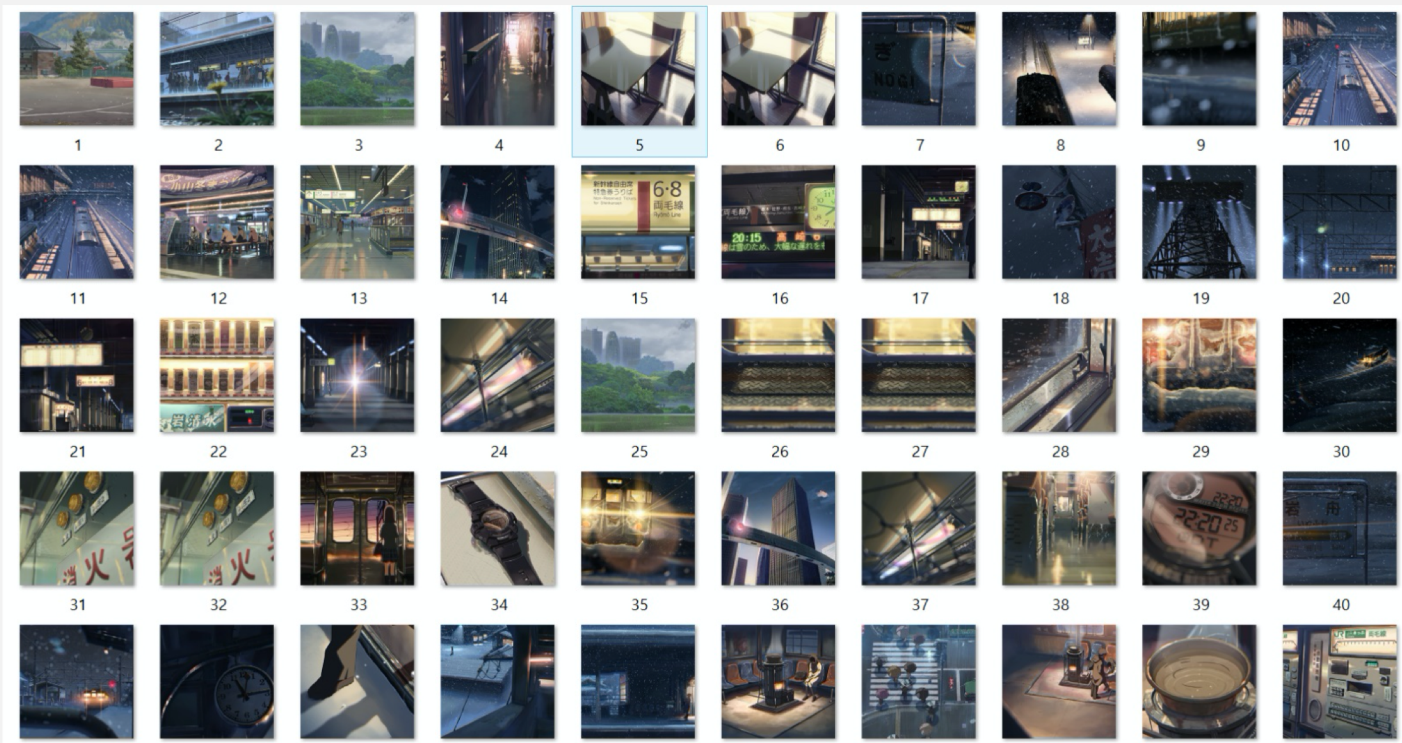
Cycle GAN의 한계

프로젝트 주제

Cycle GAN 구현

Cycle GAN 구현하기

1. 데이터 구하기 (신카이 마코토 감독 영화 캡쳐본)



Cycle GAN 구현하기

2. 전처리 하기

포토샵 자동처리 사용

3. 학습시키기

https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix 참고

Cycle GAN 구현하기

4. Test (신카이 마코토 스타일)

Cycle GAN 구현하기

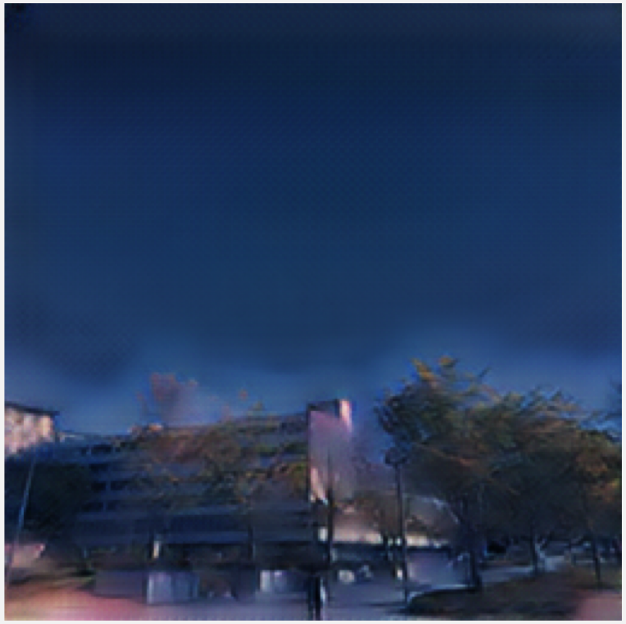
4. Test (신카이 마코토 스타일)

Cycle GAN 구현하기

4. Test (신카이 마코토 스타일)

Cycle GAN 구현하기

4. Test (신카이 마코토 스타일)



Cycle GAN 구현하기

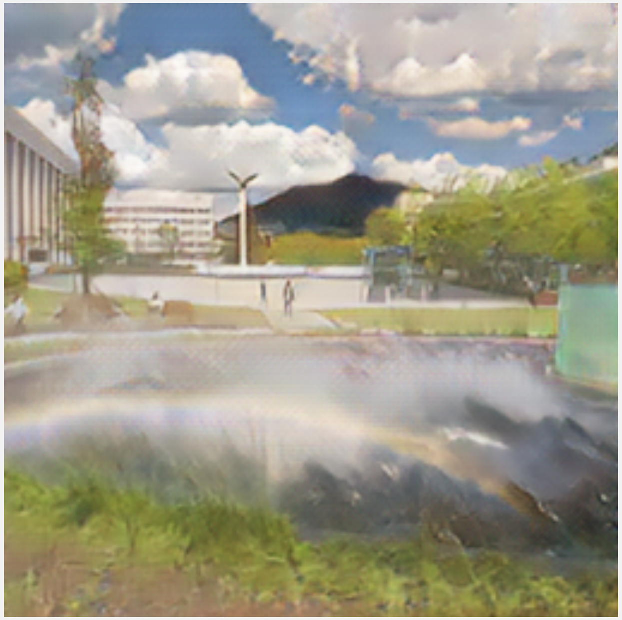
4. Test (신카이 마코토 스타일)

Cycle GAN 구현하기

4. Test (신카이 마코토 스타일)

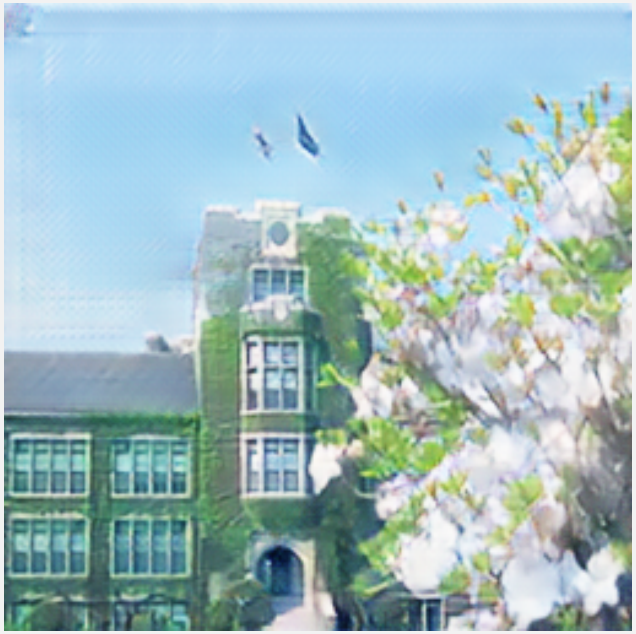
Cycle GAN 구현하기

4. Test (신카이 마코토 스타일)



Cycle GAN 구현하기

4. Test (신카이 마코토 스타일)

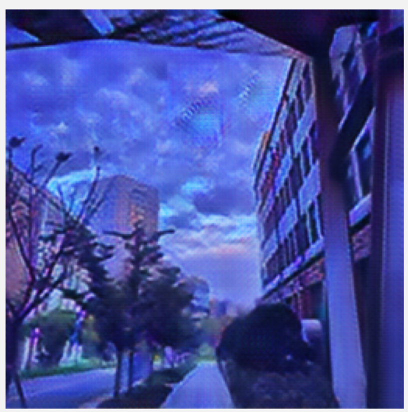


Cycle GAN 구현하기

4. Test (겨울왕국 스타일)





Cycle GAN 구현하기

4. Test (인물사진 →고전 명화 스타일)






Cycle GAN 구현하기

5. 구현 한계

a. data set 의 분포가 고르지 않다.

ex) 겨울 왕국의 경우 밤의 사진이 많아 겨울 왕국 이미지를 대표하지 못함

b. 시간이 부족해서 충분히 학습시키기 어려웠다.

c. 양쪽 도메인 사진의 피사체에 차이가 컸다.

부록

CartoonGAN?

사진을 만화의 그림체로 변형하는데 특화된 GAN 모델.

1. CycleGAN이 bidirectional mapping를 하는데 반해, CartoonGAN은 backward mapping가 필요없다. Cycle GAN 보다 30 ~ 50 % 빠르다

2. L1 Loss대신 Content Loss 사용.

$$\mathcal{L}_{con}(G,D) = \mathbb{E}_{x\sim p_{data}(x)}\left[\left\|VGG\big(G(x)\big) - VGG(x)\right\|_1\right]$$

3. source data의 edge를 흐릿하게 만들어 Loss에 추가함.

$$\mathcal{L}_{edge}(D) = \mathbb{E}_{e\sim p_{data}(e)}[\log(1 - D(e)]$$

즉, 흐릿한 edge의 data에 대해 패널티를 부과해 Generator가 선명한 edge를 가진 data를 생성하도록 유도

4. GAN을 학습시키기 전에 VGG Network를 먼저 학습시킴. 자세한 설명은 생략 …

+CartoonGAN

Test (신카이 마코토 스타일)

완성도가 만족스럽지 않음

→ Discriminator의 학습사진들이 test 사진과 차이가 많이 나기 때문?

결과 비교

- 학습 속도

Cartoon GAN **Cycle GAN**

같은 epoch을 학습하는데 Cartoon GAN이 Cycle GAN보다 확실히 빨랐다.

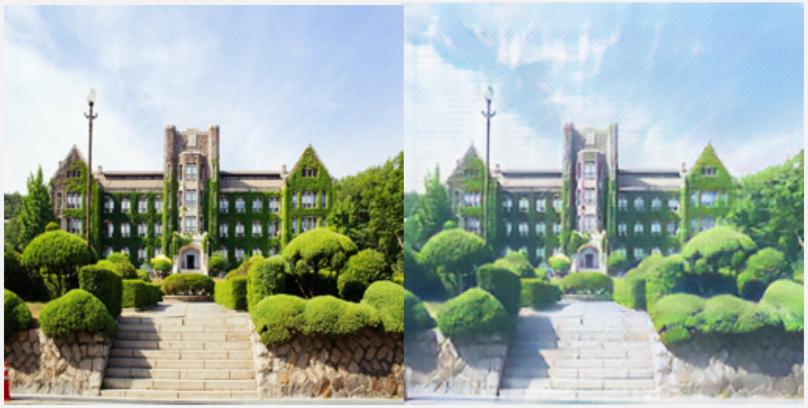
결과 비교

- 성능

Cartoon GAN　　Cycle GAN

같은 input 데이터로 학습시켰을 때 Cycle GAN의 성능이 훨씬 좋았다.

Cartoon GAN



Cycle GAN

참고 문헌

[1] 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).

[2] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).

[3] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

[4] Chen, Y., Lai, Y. K., & Liu, Y. J. (2018). Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9465-9474).

감사합니다