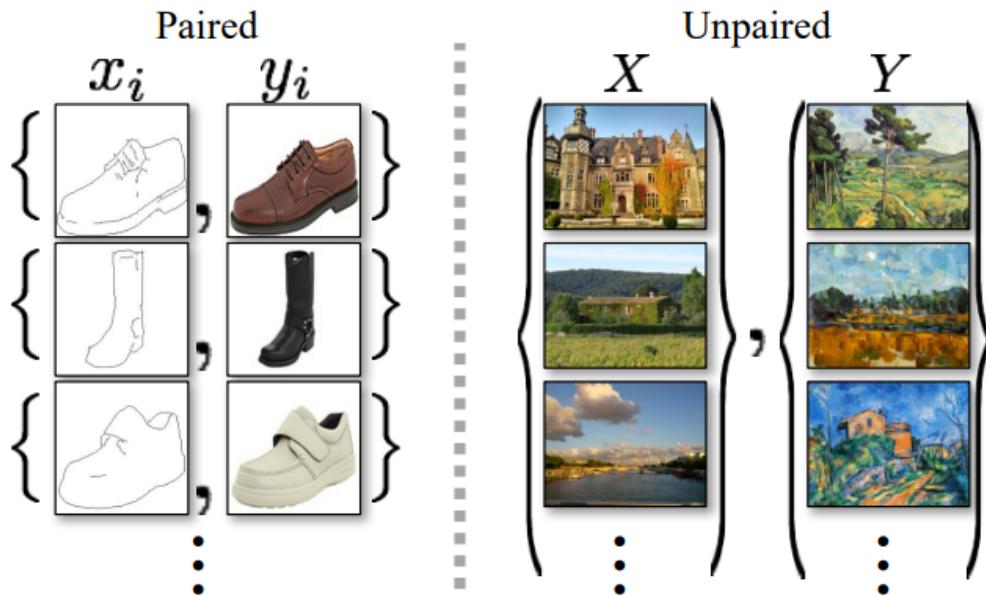


# CycleGAN : Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

## Pix2Pix의 한계점

- Pix2Pix는 서로 다른 두 도메인 X,Y의 데이터 두개를 한쌍으로 묶어서 학습을 진행
  - colorization과 같은 태스크에서는 데이터 셋을 구성하기 쉬우나 그렇지 않은 경우가 대부분



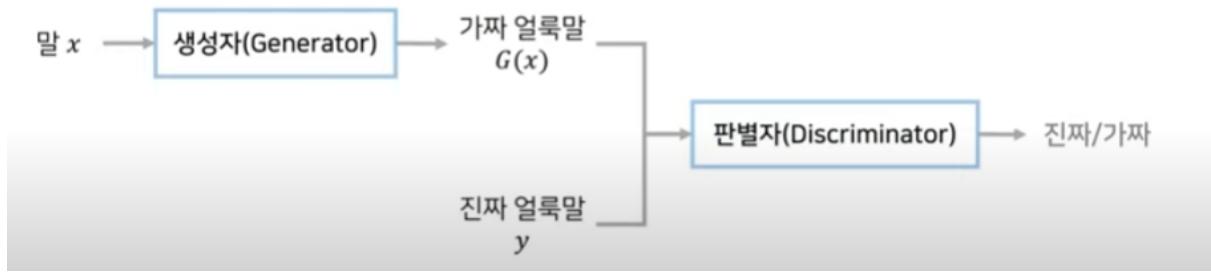
→ 한쌍을 묶지 않은 Unpaired 데이터 셋에서도 가능하지 않을까 → Cycle GAN을 제안

- 특정한 이미지가 x가 주어졌을 때, target domain Y의 그럴싸한 이미지로 바꾸도록 학습할 수 있음

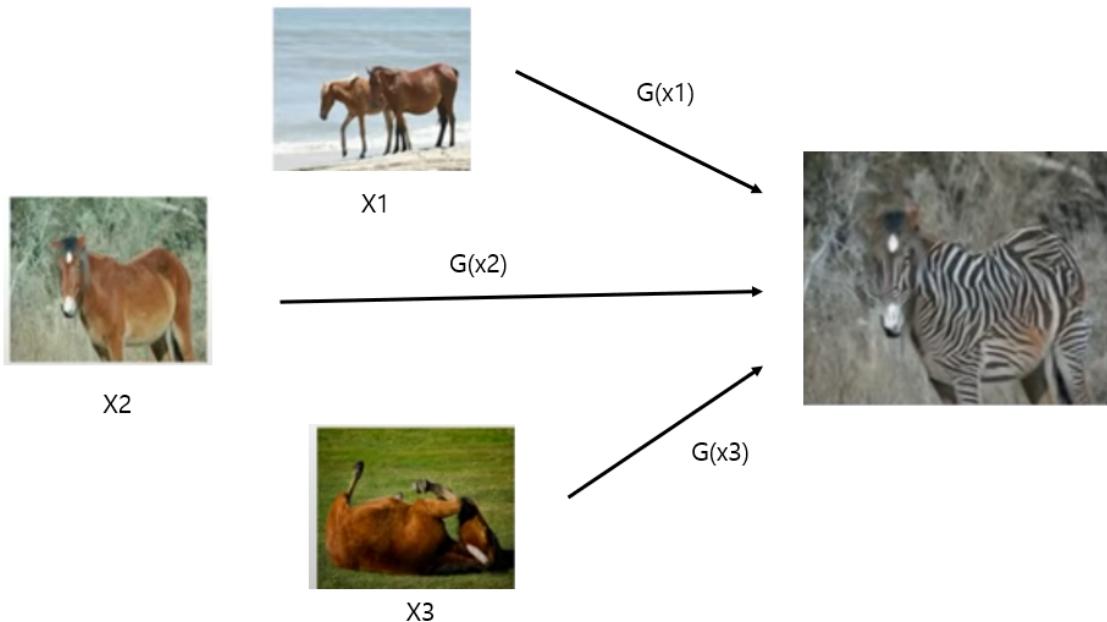
$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = & \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D_Y(y)] \\ & + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log(1 - D_Y(G(x)))]\end{aligned}$$

- 기존 Gan loss

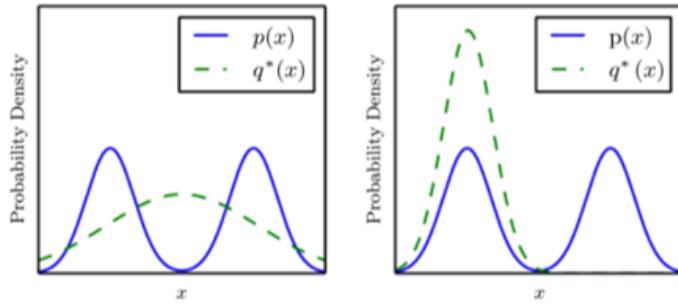
- $x$ 와  $y$ 가 서로 매칭되지 않았기 때문에, 들어온  $x$ 데이터에 대해 그럴싸 하기만 한다면  $y$ 의 어떤 이미지가 나와도 문제를 해결했다고 생각함



- 하지만 이 및  $x$ 의 Content를 유지한 상태로 translation이 가능하다는 보장이 없음  
→ 즉, 말을 얼룩말로 바꿀때 가짜 얼룩말을 바꾸는 것이 목표지만, 서로 매칭 되어 있지 않음



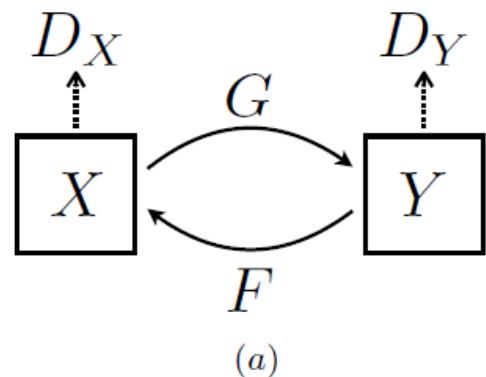
# Mode collapse 발생



- Mode collapse라는 문제는 이렇게 파란색의 실제 데이터의 분포가 주어졌을 때, 우리는 generator가 이 실제 데이터의 분포와 최대한 유사하게 학습하기를 바라게 됨.
- 그러나 단순히 loss만을 줄이기 위해서 학습을 하기 때문에 G가 이렇게 전체 데이터 분포를 찾지 못하고 오른쪽 그림과 같이 한번에 하나의 mode에만 강하게 몰리게 되는 경우가 발생을하게 됨.
- 이렇게 되면 서로 다른 두 그림의 아웃풋이 이렇게 동일한 사진이 나오는 경우가 발생함
- 매칭되는  $y_1$  없이 단순한 입력 이미지  $x_1$ 의 특성을 타겟 도메인  $Y$ 의 특성으로 바꾸어 보고자 함
- 이때 GAN Loss만 사용하면,  $G$ 는 어떤 입력이든  $Y$  도메인에 해당하는 하나의 이미지만 제시 할 수 있음
  - 판별자 입장에서는 있을법한  $Y$  도메인의 이미지로 적절히 분류됨
  - 다시 말하면,  $x_1$ 의 content정보를 아예 변경 시킬 수 있음
  - 따라서 추가적인 제약이 필요함 → 추가적인 Loss가 필요함

## CycleGAN

- CycleGAN은  $G(x)$ 가 다시 원본 이미지  $x$ 로 재구성(reconstruct)될 수 있도록 함
  - 원본 이미지의 content는 보존하고 도메인과 관련된 특성만 바꿔보도록 하자
- 이를 위해 2가지 변환기(translator)를 사용함
  - $G: X \rightarrow Y$
  - $F: Y \rightarrow X$ 
    - $G$ 와  $F$ 는 서로 역함수 관계임



- 목표

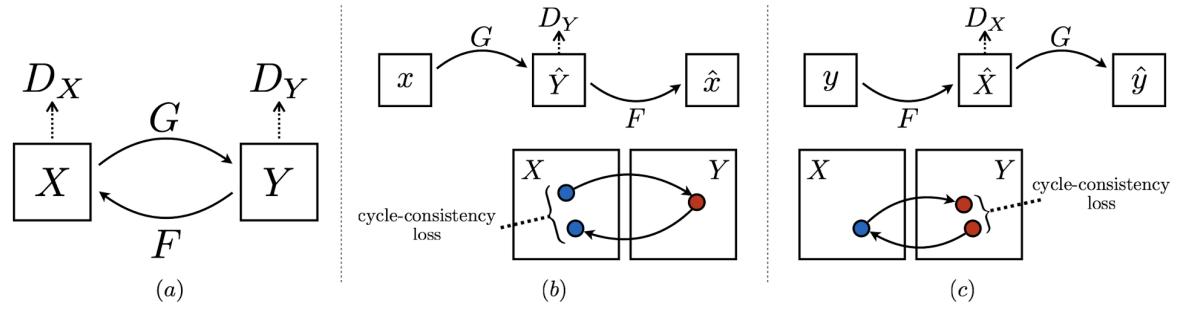
$$F(G(x)) \approx x \text{ and } G(F(y)) \approx y$$

- 추가적인 조건으로 Cycle-Consistency loss를 사용

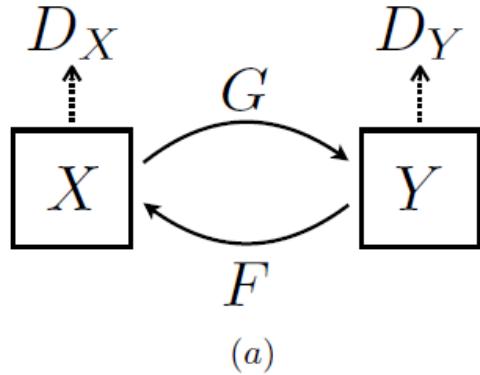


## CycleGAN 전체 목적 함수

- 한 쌍으로 묶이지 않은(Unpaired) 데이터를 학습하기 위하여 Cycle loss를 사용

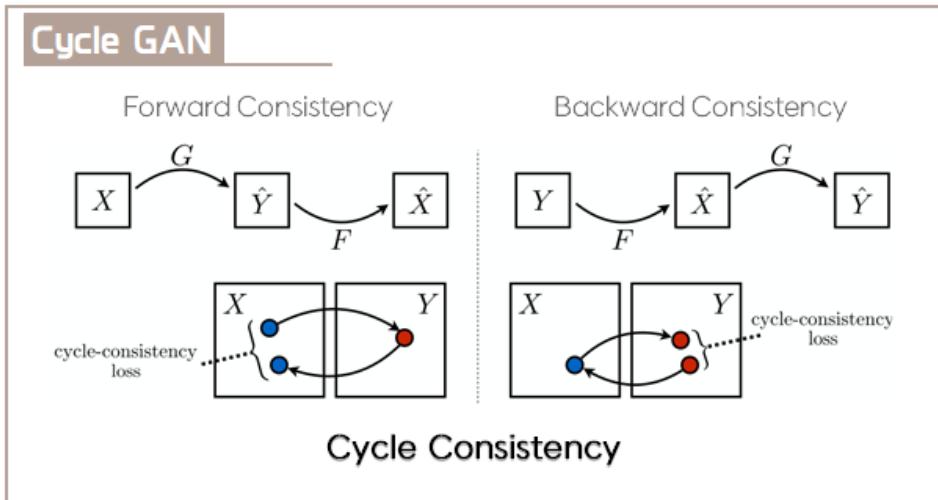


▼ A 그림



- $x$ 는  $X$ 에 속하는,  $y$ 는  $Y$ 에 속하는 샘플
- $x \sim p_{data}(x), y \sim p_{data}(y)$
- 위의 그림 (a)와 같이 우리의 모델은 두 개의 매핑 함수  $G:X \rightarrow Y$ 와  $F:Y \rightarrow X$ 를 포함
- 추가로 저자는 두 개의 적대적인(adversarial) discriminator  $D_X$ 와  $D_Y$ 를 도입
- 적함수(objective)는 adversarial losses와 cycle consistency losses, 두 종류의 항으로 구성되어 있음

▼ B,C 그림



- 즉 그림에서 사진으로 맵핑하는 동작 과정을 forward consistency
- 반대의 과정을 backward consistency라고 함
- 이때 Forward와 backward 각각 모두 generator를 거쳐서 한바퀴를 돌아오면 다시 처음 자기자리로 돌아와야 하기 때문에 '순환 일관성(Cycle Consistency)'라는 이름이 붙었고, cycle GAN에서는 이러한 원리를 가져와서 손실함수를 만드는 것

## 전체 Loss

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = & \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ & + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ & + \underline{\lambda} \mathcal{L}_{\text{cyc}}(G, F),\end{aligned}$$

## GAN Loss

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

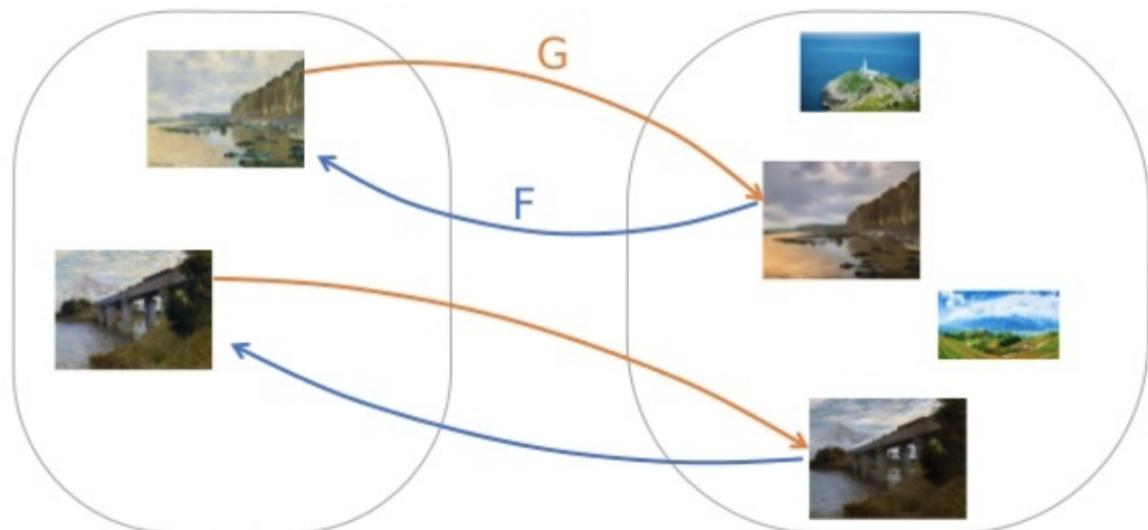
▼ 수식 설명

- $G : X \rightarrow Y$  translator,  $D_y : Y$  discriminator
- $y \sim p_{\text{data}}(y)$ 는  $Y$ 의 data 분포를 따르는 원소  $y$ 를 말하고  $D_y$ 는  $Y$ 의 데이터 분포에서 있는지( $Y$ 도메인인지) 아닌지 판단하는 discriminator로 0~1 사이의 확률 값을 반환
- $Y$  분포에서 있다고 판단하면 1에 가까운 값을 아니면 0에 가까운 값이 나옴. 이것들의 평균이 loss로 사용
- $D_y$ 가 잘 판단한다면 두 항이 0에 가까워져 GAN loss는 0에 가까워지게 됨. 반대로 잘못 판단하면 음수 값이 나옴
- $\min_G \min_{D_y} L_{GAN}(G, D_y, Y, X)$ 로 나타낼 수 있음
- $F : Y \rightarrow X$ 와  $D_y$  대해서도 유사한 adversarial loss를 적용합니다
- 이는  $\min_F \min_{D_x} L_{GAN}(F, D_x, Y, X)$

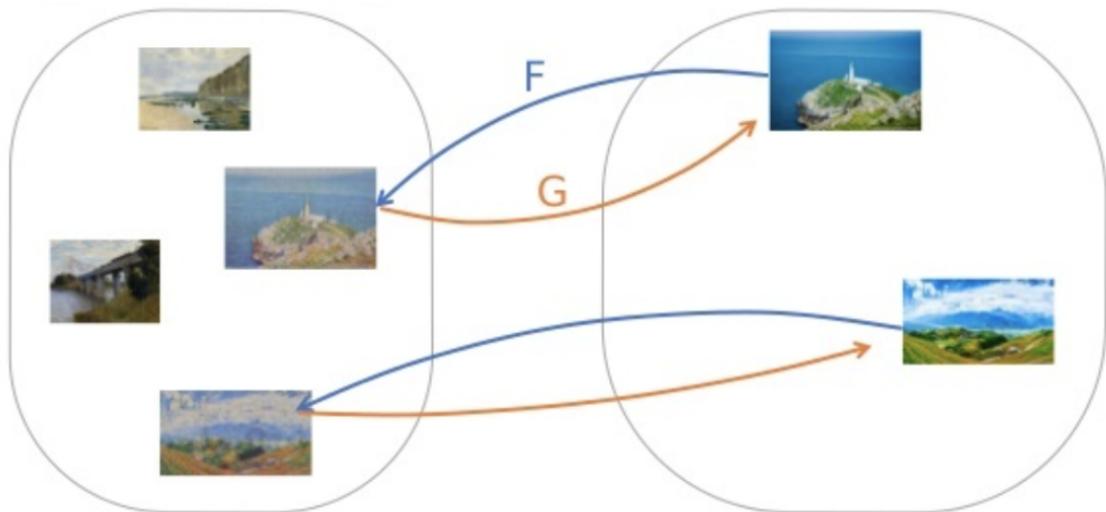
⇒ Adversarial loss : target domain에 있을 법한 이미지를 생성하는 loss

## Cycle Loss

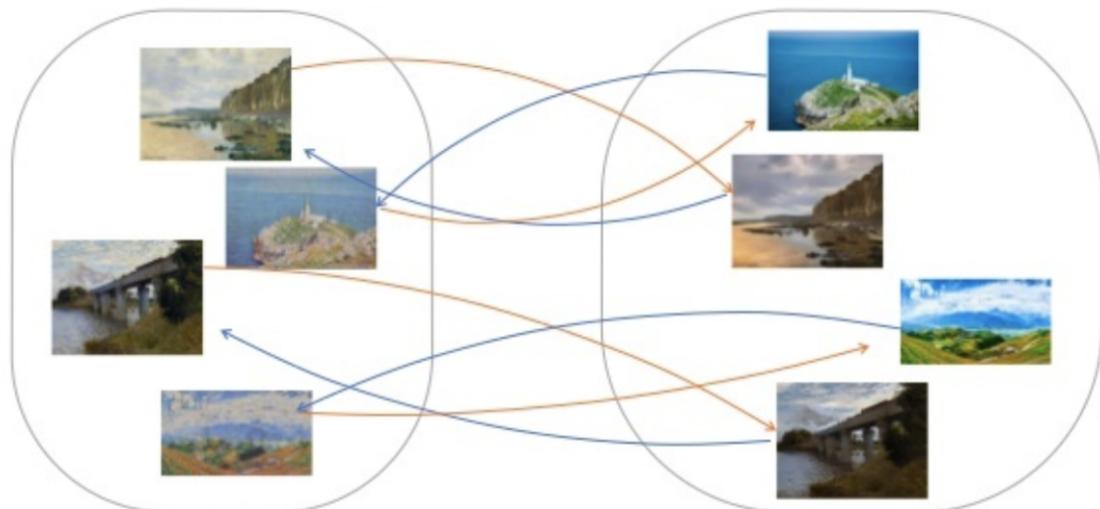
$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = & \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|_1] \\ & + \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|_1]\end{aligned}$$



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



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



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

- forward cycle consistency

$$x \rightarrow G(x) \rightarrow F(G(x)) \approx x$$

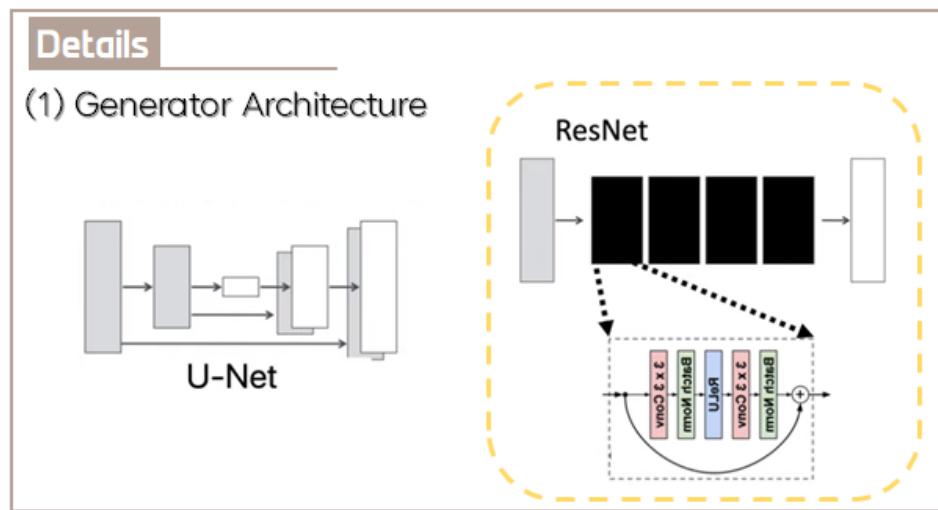
- backward cycle consistency

$$y \rightarrow F(y) \rightarrow G(F(y)) \approx y$$

⇒ Cycle-consistent loss: 입력과 매칭되는 image-to-image translation 결과 이미지를 찾을 수 있도록 함

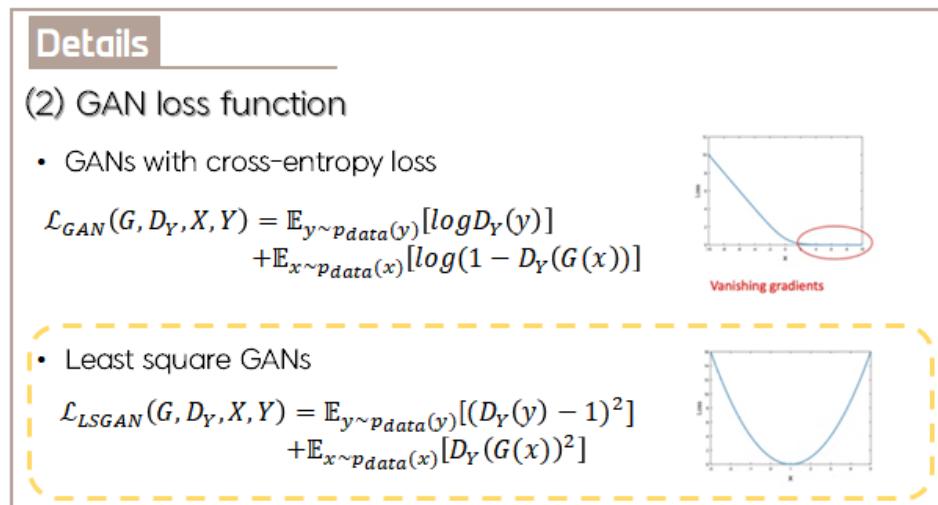
## Cycle GAN 구현

### 네트워크 아키텍쳐



- Residual Block을 활용하는 아키텍쳐 및 instance normalization을 활용
- 이미지 내 패치 단위로 진위 여부를 판별하는 판별자를 사용

### 학습 방법



## Least Squares Loss

Discriminator Loss	$\mathbb{E}_{\mathbf{x}}[(D(\mathbf{x}) - 1)^2] + \mathbb{E}_{\mathbf{z}}[(D(G(\mathbf{z})))^2]$
Generator Loss	$\mathbb{E}_{\mathbf{z}}[(D(G(\mathbf{z})) - 1)^2]$

- 
- Least-squares-loss: 기존의 cross-entropy 기반의 loss 대신에 MSE 기반의 loss를 사용
    - 실제 이미지 분포와 더욱 가까운 이미지를 생성 할 수 있으며, 학습이 안정화 됨
  - Replay Buffer: 생성자가 만든 이전 50개의 이미지를 저장해 두고, 이를 이용해 판별자를 업데이트
    - 모델의 Oscillation을 개선

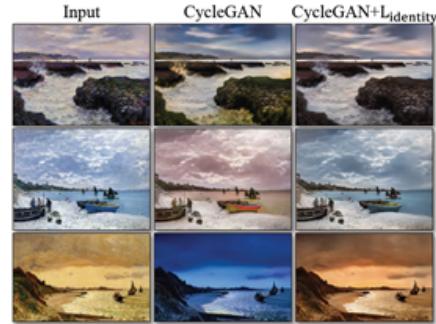
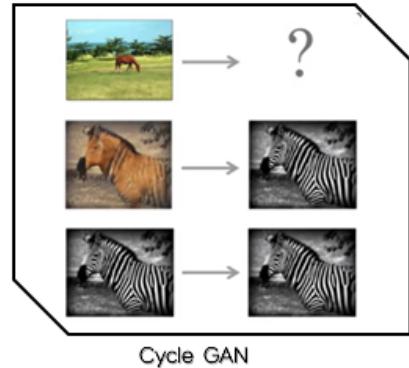
## Identity Loss

- 색상 구성을 보존해야 할 때는 identity loss를 추가적으로 사용 할 수 있음
  - 그림을 사진으로 변경할 때처럼 색상 정보가 유지되어야 하는 task에서 효과적으로 사용 할 수 있음

$$L_{identity}(G, F) = E_{y \sim p_{data}(y)}[\| G(y) - y \|_1] + E_{x \sim p_{data}(x)}[\| F(x) - x \|_1]$$

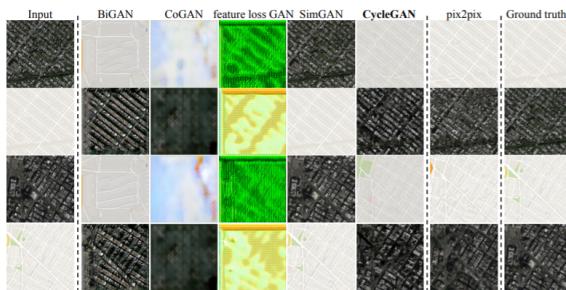
## Details

### (3) Identity loss function



## CycleGAN 결과

- 이전 연구에 비해 우수한 생성 결과를 확인 할 수 있음
- Pix2Pix에 비교할 만한 점수가 나옴
  - pix2pix는 pair된 dataset을 이용해 학습한 결과



Loss	Per-pixel acc.	Per-class acc.	Class IOU
Cycle alone	0.22	0.07	0.02
GAN alone	0.51	0.11	0.08
GAN + forward cycle	<b>0.55</b>	<b>0.18</b>	<b>0.12</b>
GAN + backward cycle	0.39	0.14	0.06
CycleGAN (ours)	0.52	0.17	0.11

Table 4: Ablation study: FCN-scores for different variants of our method, evaluated on Cityscapes labels→photo.

Loss	Per-pixel acc.	Per-class acc.	Class IOU
Cycle alone	0.10	0.05	0.02
GAN alone	0.53	0.11	0.07
GAN + forward cycle	0.49	0.11	0.07
GAN + backward cycle	0.01	0.06	0.01
CycleGAN (ours)	<b>0.58</b>	<b>0.22</b>	<b>0.16</b>

Table 5: Ablation study: classification performance of photo→labels for different losses, evaluated on Cityscapes.

- CycleGAN의 손실 함수를 모두 사용 했을 때 가장 우수한 결과가 나옴.
  - GAN alone과 GAN +forward는 평가 지표상으로는 높을 수 있으나 model collapse의 문제

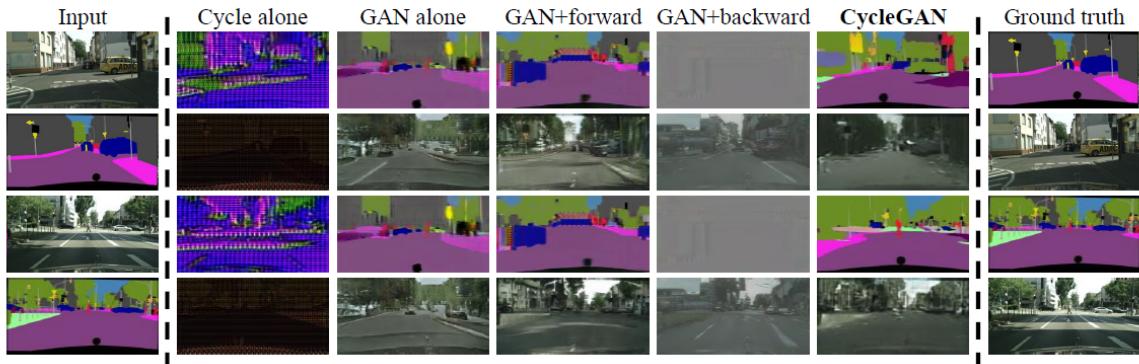
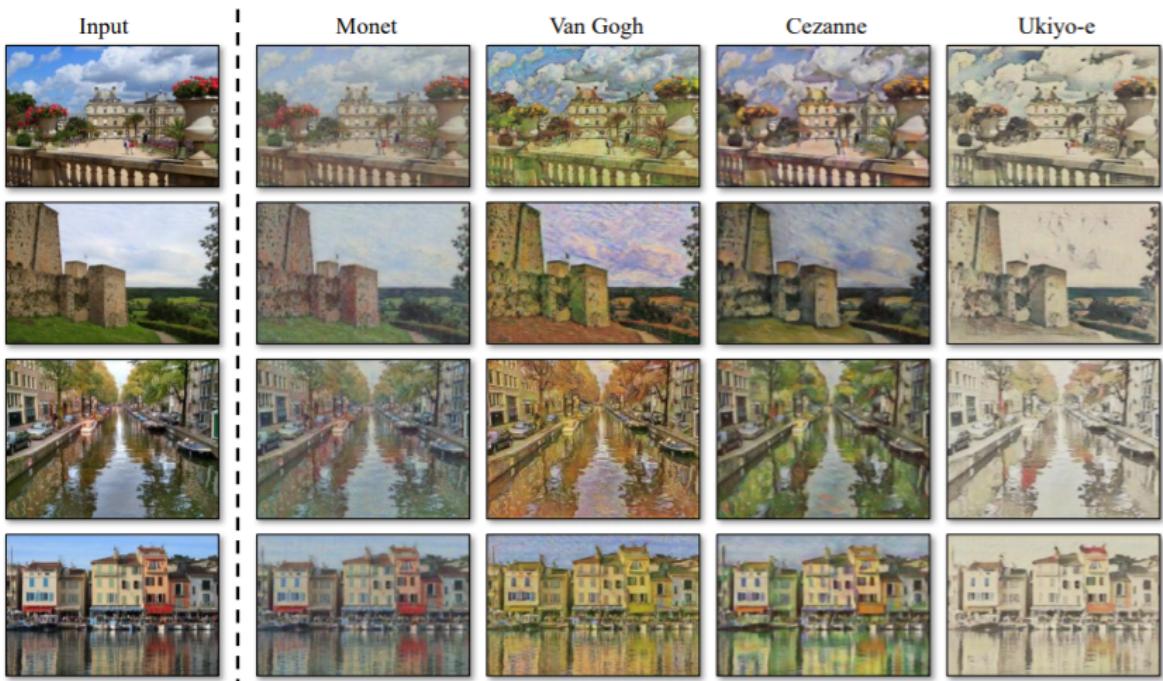
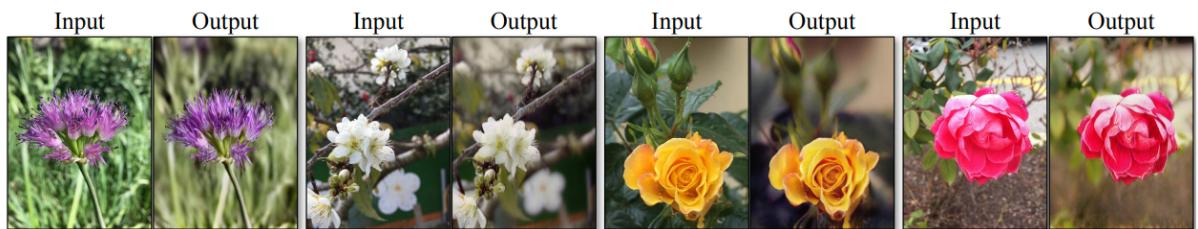


Figure 7: Different variants of our method for mapping labels $\leftrightarrow$ photos trained on cityscapes. From left to right: input, cycle-consistency loss alone, adversarial loss alone, GAN + forward cycle-consistency loss ( $F(G(x)) \approx x$ ), GAN + backward cycle-consistency loss ( $G(F(y)) \approx y$ ), CycleGAN (our full method), and ground truth. Both *Cycle alone* and *GAN + backward* fail to produce images similar to the target domain. *GAN alone* and *GAN + forward* suffer from mode collapse, producing identical label maps regardless of the input photo.

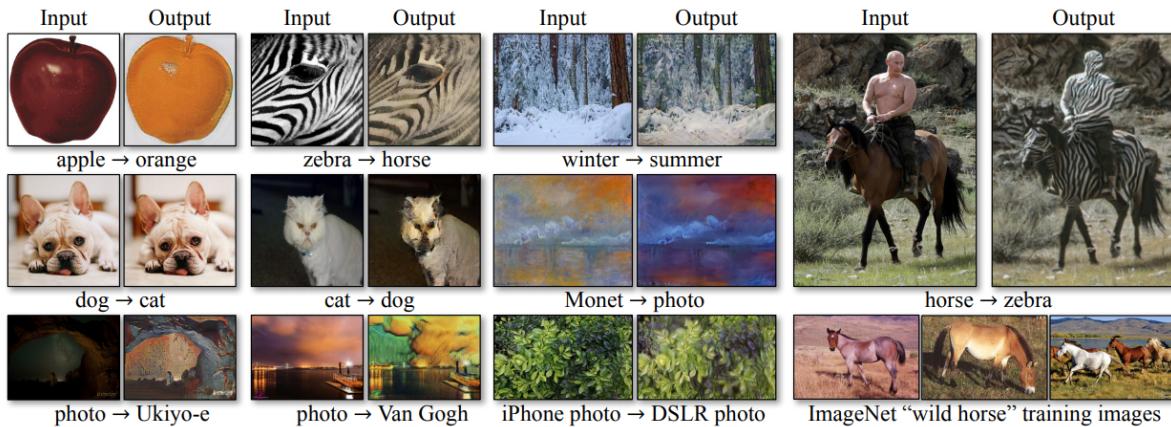
- Style transfer



- 스마트폰 촬영을 DSLR처럼 변환 가능



- Limitations and Discussion



## 참고 링크

- 블로그

### CycleGAN-Unpaired 데이터를 학습하고 이미지 변환하기

이번 포스팅에서는 이미 너무나도 유명한 CycleGAN 논문을 리뷰해보고자 한다. 본 글의 구성은 첫 째로 CycleGAN을 접한 저자에 대해 소개하는 것으로 시작한다. 논문 저자는 CycleGAN 이후로도 2020년 현재까지 이미지 생성 모델 분야에서 활발한 연구 활동을 하고 있고 주제가 흥미롭다. <https://medium.com/curg/cyclegan-unpaired-%EB%8D%B0%EC%9D%B4%ED%84% B0%EB%A5%BC-%ED%95%99%EC%8A%B5%ED%95%98%EA%B3%A0-%EC%9D% B4%EB%AF%8B%EC%A7%80-%EB%B3%80%ED%99%98%ED%95%98%EA%B8% B>

<http://www.youngdata-study.tistory.com/9> <https://bellzero.tistory.com/26>

<https://dambi-ml.tistory.com/7> <https://yun905.tistory.com/22>

<https://comlini8-8.tistory.com/9>

<https://velog.io/@chy0428/CycleGAN>

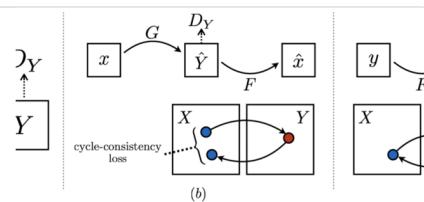
<https://velog.io/@tobigs-gm1/Image-to-Image-Translation>

- 유튜브

### CycleGAN - 딥러닝 기반의 이미지 변환 기법 [꼼꼼한 딥러닝 논문 리뷰와 코드 실습]

오늘 영상에서 다루는 논문은 CycleGAN (ICCV 2017)입니다. GAN을 기반으로 하여 Image-to-Image Translation 네트워크를 제안한 논문으로 잘 알려져 있습니다. 본 논문은 Pix2Pix와 비교했을 때, 쌍으로 구성되지 않은 unpaired data를 사용하여 다양한 task에서 위상난 성능을 보입니다.

<https://www.youtube.com/watch?v=dr9Yf8EY4J4&list=PLRx0vPvlEmdADpce8aoBhNnDaaHQN1Typ&index=34>



- 깃허브

### [GAN] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

CycleGAN 논문 구현 및 생각 과정과 정리 현재 하고 있는 task에서 cycleGAN을 활용한 방법이 있어 CycleGAN 부터 제대로 알고 구현하고자 좀 딥하게 읽고 생각해봤습니다. CycleGAN 논문과 공식 구현 코드입니다. CycleGAN 이해에 도움이 될만한 자료입니다. 논문의 공동 저자인 박태성님의 한글 발표 자료입니다.  
<https://subinium.github.io/CycleGAN/>



### PyTorch-CycleGAN/train at master · aitorzip/PyTorch-CycleGAN

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

🔗 <https://github.com/aitorzip/PyTorch-CycleGAN/blob/master/train>

### aitorzip/PyTorch-CycleGAN



A clean and readable Pytorch implementation of CycleGAN

1 Contributor 29 Issues 843 Stars 237 Forks