Generative Adversarial Network
: Image-to-Image Translation - CycleGAN

Seokkyeong Boo

Image-to-Image Translation : Cycle GAN

- image-to-image Translation

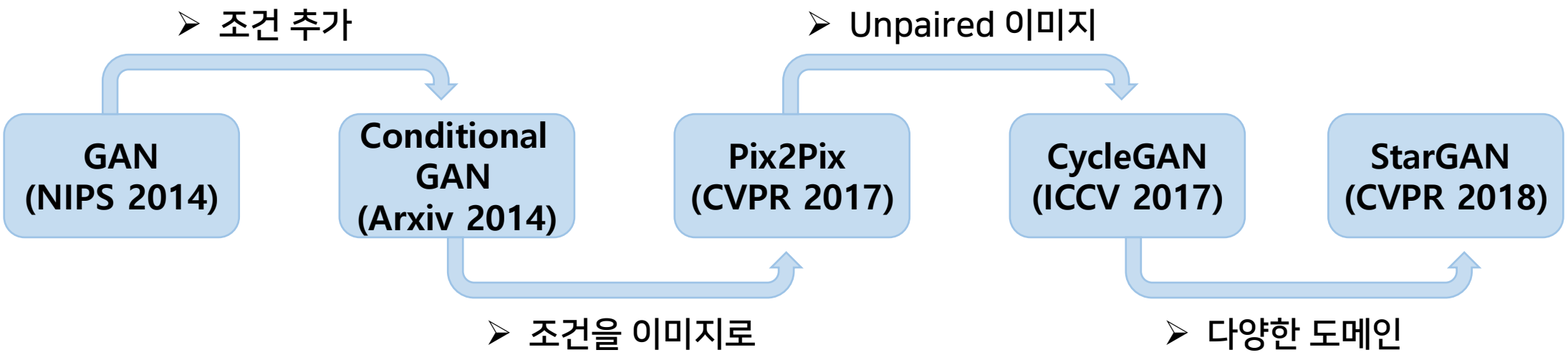


Image-to-Image Translation : Cycle GAN

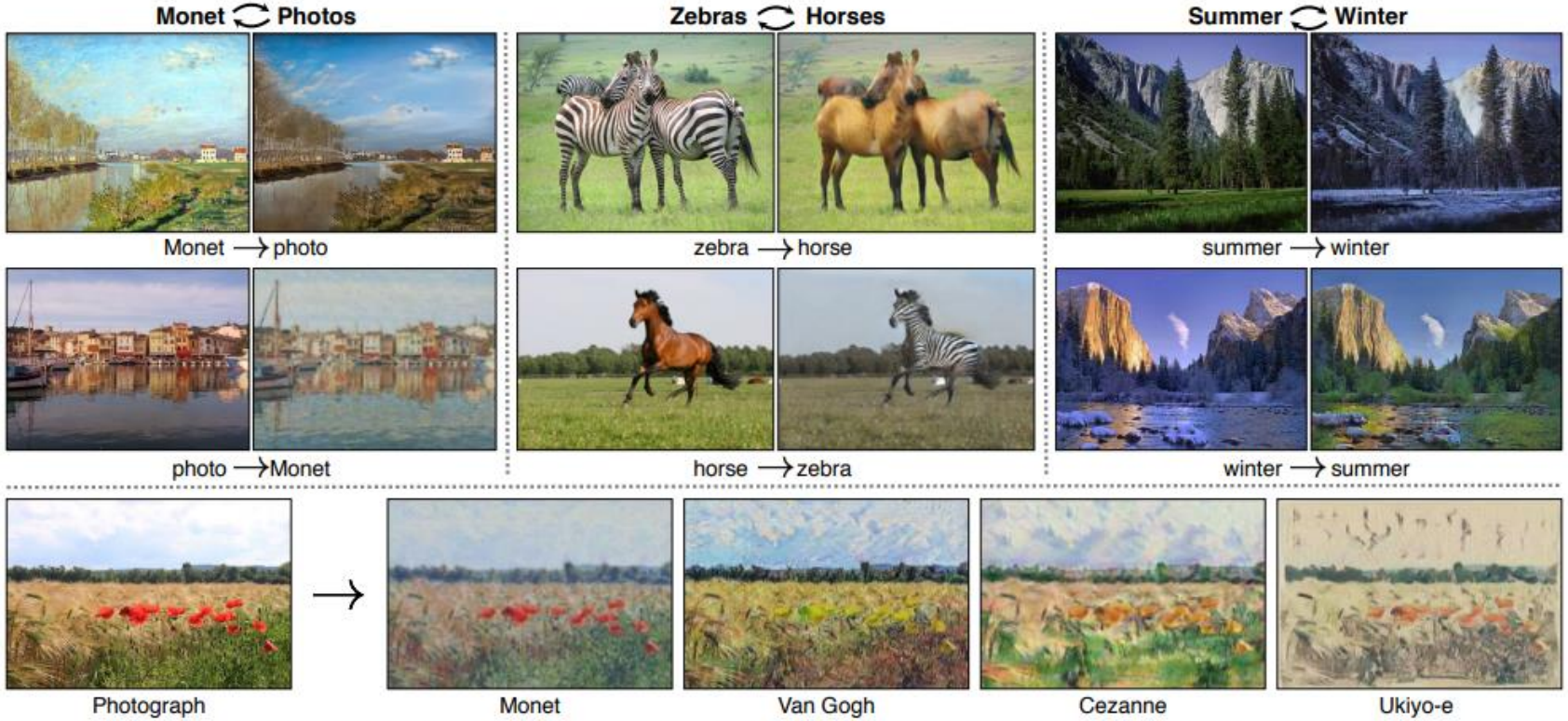
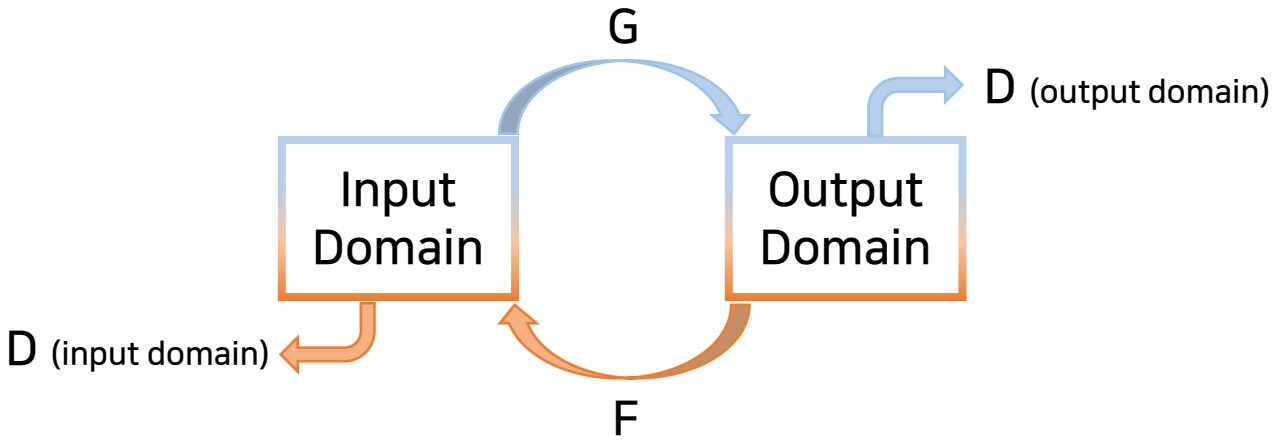


Figure 1: Given any two unordered image collections X and Y, our algorithm learns to automatically "translate" an image from one into the other and vice versa: *(left)* Monet paintings and landscape photos from Flickr; *(center)* zebras and horses from ImageNet; *(right)* summer and winter Yosemite photos from Flickr. Example application *(bottom)*: using a collection of paintings of famous artists, our method learns to render natural photographs into the respective styles.

Image-to-Image Translation : Cycle GAN

- Cycle GAN
  - 생성자는 만들어낸 이미지는 다시 입력 이미지로 재구성될 수 있도록 한다.
  - 입력 이미지의 구성을 보존하면서 변환할 도메인과 관련된 특성으로 변환한다.
  - 따라서, 2개의 변환기를 사용한다.



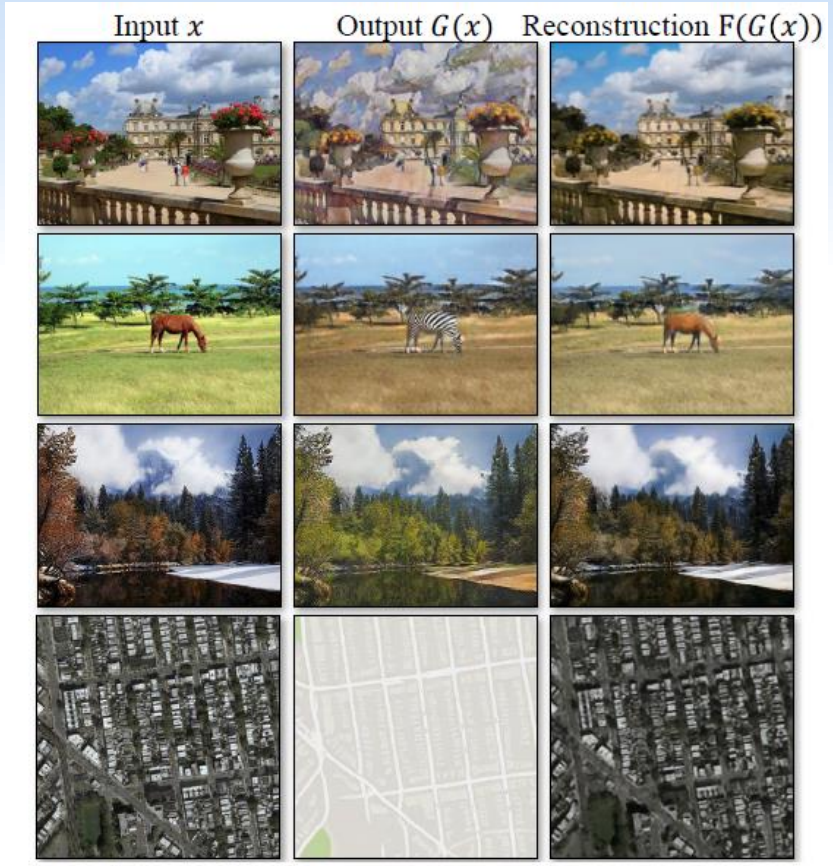


Figure 4: The input images x, output images $G(x)$ and the reconstructed images $F(G(x))$ from various experiments. From top to bottom: photo↔Cezanne, horses↔zebras, winter→summer Yosemite, aerial photos↔Google maps.

Image-to-Image Translation : Cycle GAN

- Adversarial Loss
  - G가 생성하는 생성 이미지 G(x)와 D(y)에 대해서는 다음과 같다.

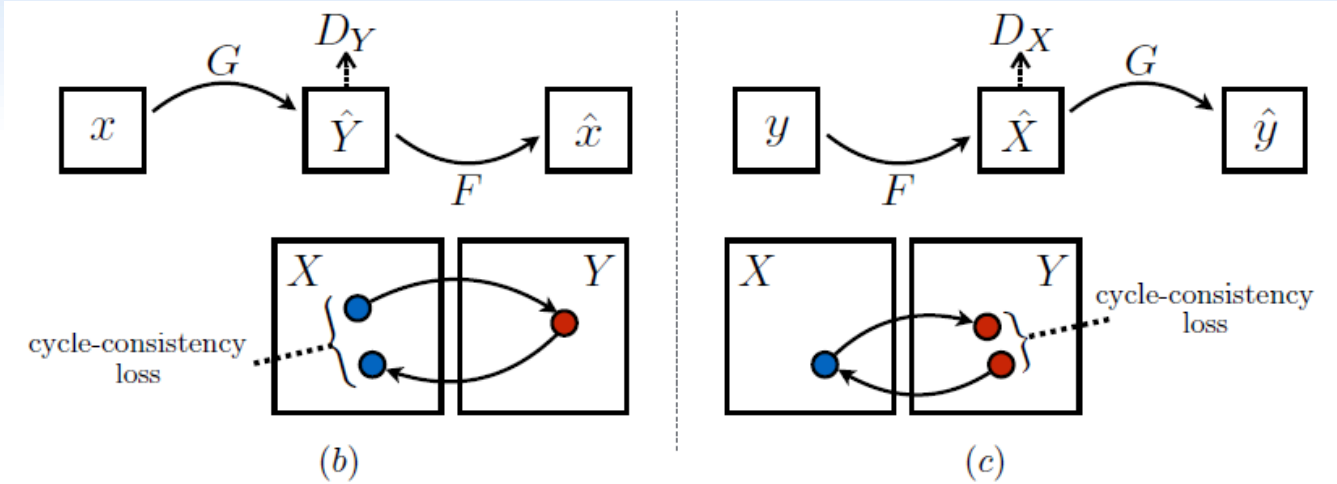
$$Loss_{GAN}\left(G, D_Y, X, Y\right) = E_{y \sim P\text{data}(y)}\left[\log D_Y\left(y\right)\right] + E_{x \sim P\text{data}(x)}\left[\log\left(1 - D_Y\left(G\left(x\right)\right)\right)\right]$$

  - F가 생성하는 이미지에 대해서도 유사한 loss를 적용한다.

$$Loss_{GAN}\left(F, D_X, X, Y\right) = E_{x \sim P\text{data}(x)}\left[\log D_X\left(x\right)\right] + E_{y \sim P\text{data}(y)}\left[\log\left(1 - D_X\left(F\left(y\right)\right)\right)\right]$$

Image-to-Image Translation : Cycle GAN

- Cycle Consistency Loss



$$Loss_{cyc}(G,F) = E_{x\sim p_{data}(x)}\left[\left\|F(G(x)) - x\right\|_1\right] + E_{y\sim p_{data}(y)}\left[\left\|G(F(y)) - y\right\|_1\right]$$

Image-to-Image Translation : Cycle GAN

➢ Full Objective

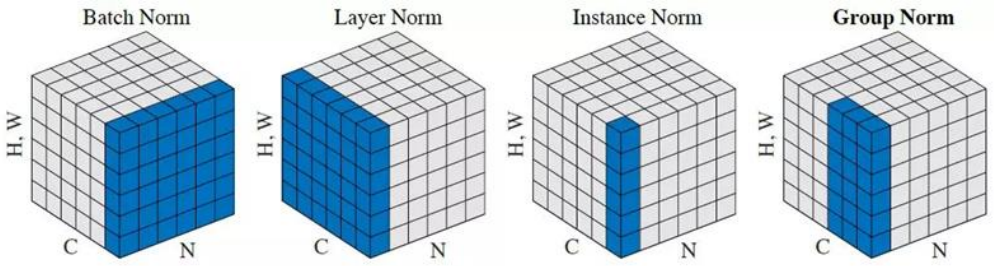
$$Loss(G, F, D_X, D_Y) = Loss_{GAN}(G, D_Y, X, Y) + Loss_{GAN}(F, D_X, X, Y) + \lambda Loss_{cyc}(G, F)$$

➢ 따라서 풀고자 하는 목표는 다음과 같다.

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

Image-to-Image Translation : Cycle GAN

- Network Architecture
  - Convolutions(stride-2) 두 번
  - 잔여 블록(Residual Block)
    - ✓ shortcut, skip connection
  - Factionally strided Convolutions 두 번
  - instance normalization
  - 판별자는 70*70 Patch GAN을 사용한다.



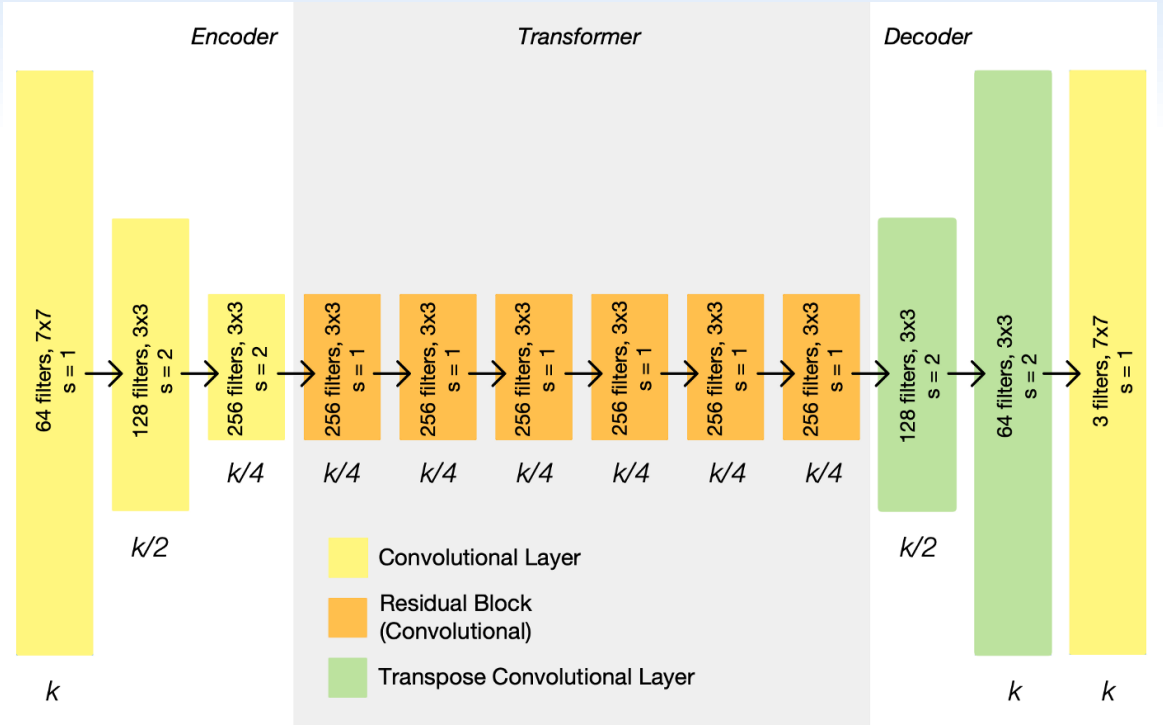


Image-to-Image Translation : Cycle GAN

- Training details
  - 논문의 모델 학습 과정 안정화 기술

  - $Loss_{GAN}$ 의 Negative log likelihood 함수를 least-squares loss로 대체했다.

$$G = E_{x \sim p_{data}(x)}\left[D\left(G\left(x\right)-1\right)^2 \right]$$

$$D = E_{y \sim p_{data}(y)}\left[\left(D\left(y\right)-1\right)^2 \right] + E_{x \sim p_{data}(x)}\left[\left(D\left(G\left(x\right)\right)\right)^2 \right]$$

  - 모델의 판별자를 최신의 생성자가 생성한 하나의 이미지를 이용하기보다 생성된 이미지 50개를 저장하여 이용했다.

  - λ는 10으로 설정했다.
  - 배치 사이즈는 1, Adam을 이용했다.
  - 처음 100 epoch에 대해서는 학습률을 0.0002를 유지, 이후 선형적으로 0에 가까워지게 학습률을 줄였다.