Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Sanghyuck Na
Nov, 27, 2020
Dongguk University
Artificial Intelligence Laboratory
shna@Dongguk.edu

0

Contents

1 Introduction

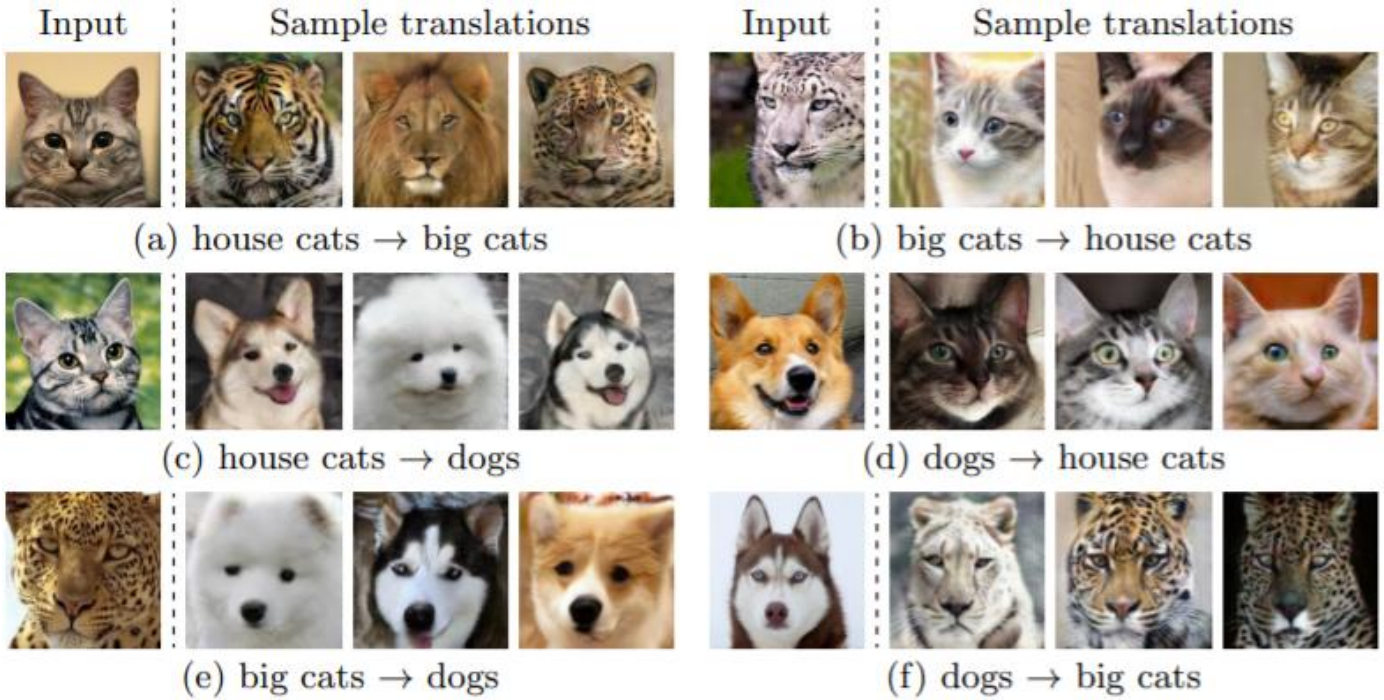
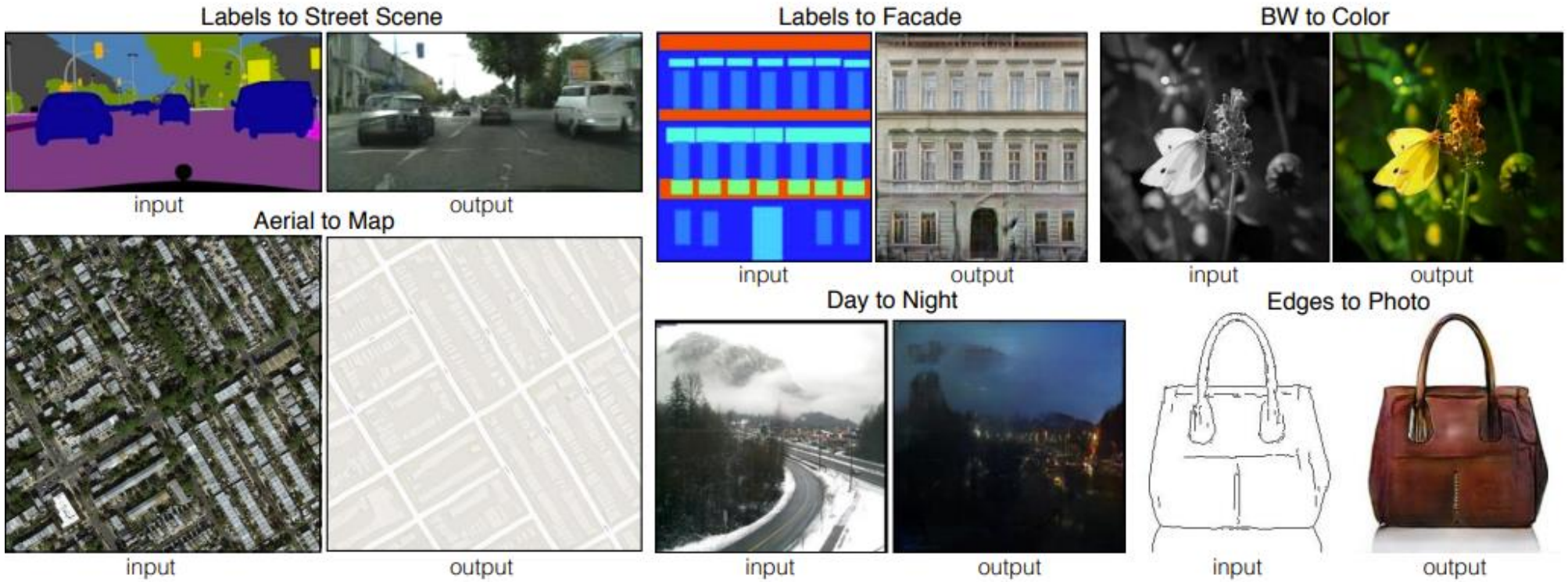


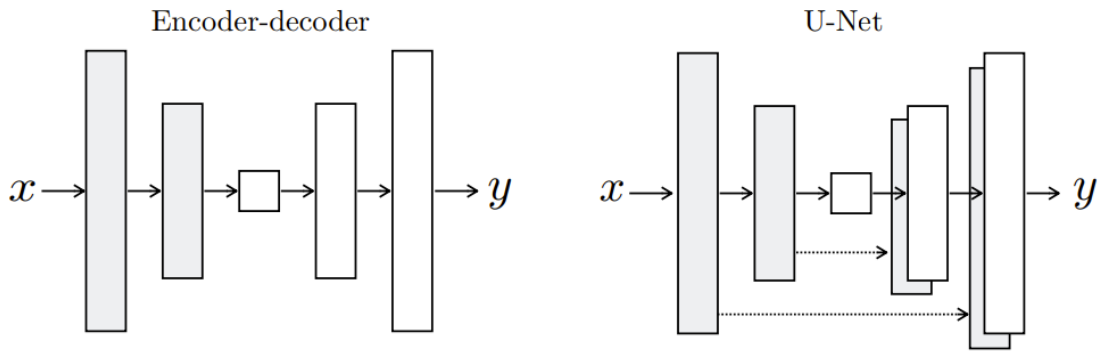
Image to image translation is to translate an image from a source domain to a target domain.

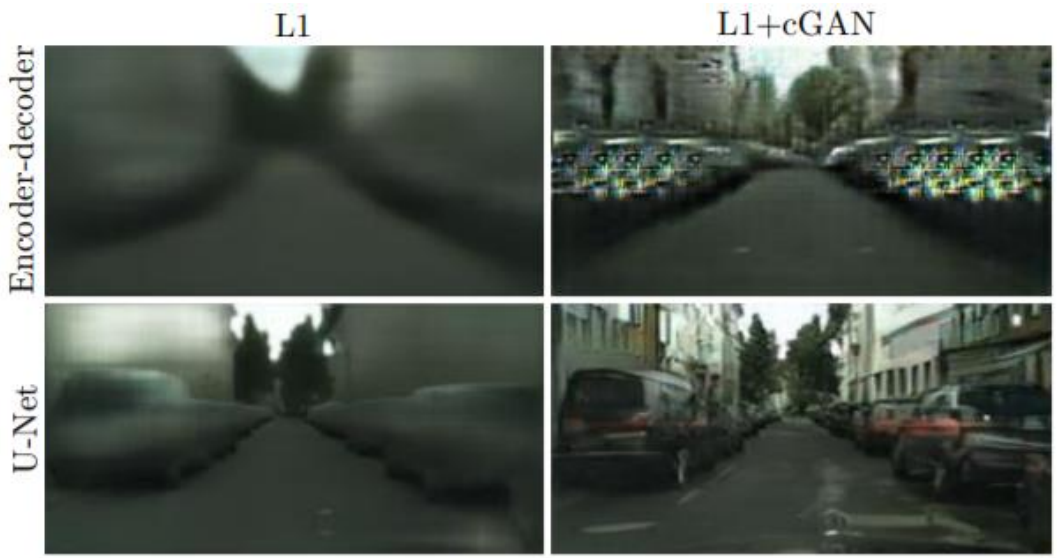
1 Introduction



It can be applied to style transfer, object transfiguration, season transfer and photo enhancement.

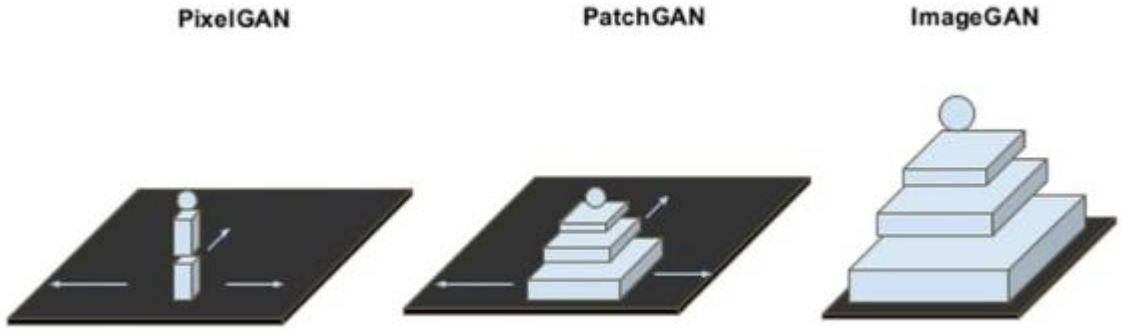
Pix2Pix

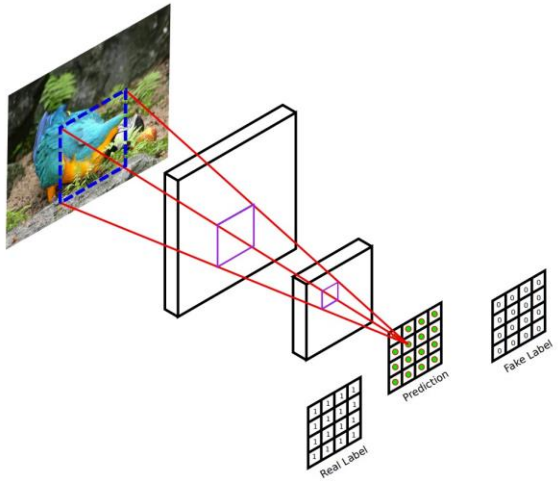




Two choices for the architecture of the generator. The "U-Net" is an encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks.

2 Pix2Pix





Unlike previous studies, PatchGAN discriminator distinguishes real or fake from patches of images, which is called markovian discriminator and Local-patch discriminator.

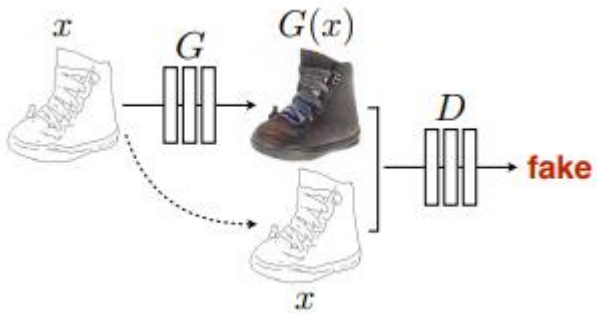
2 Pix2Pix

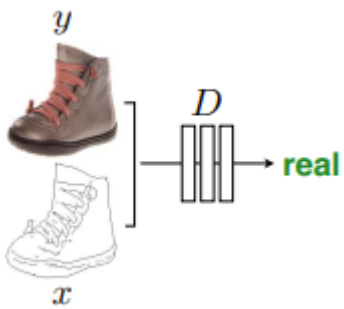
$$L_{cGAN}(G,D) = \mathbb{E}_{x,y}[logD(x,y)] + \mathbb{E}_{x,z}[\log(1 - D\big(x, G(x,z)\big)]$$

$$L_{GAN}(G,D) = \mathbb{E}_{y}[\log(D(y)] + \mathbb{E}_{x,z}[\log(1 - D\big(G(x,z)\big)]$$

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

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





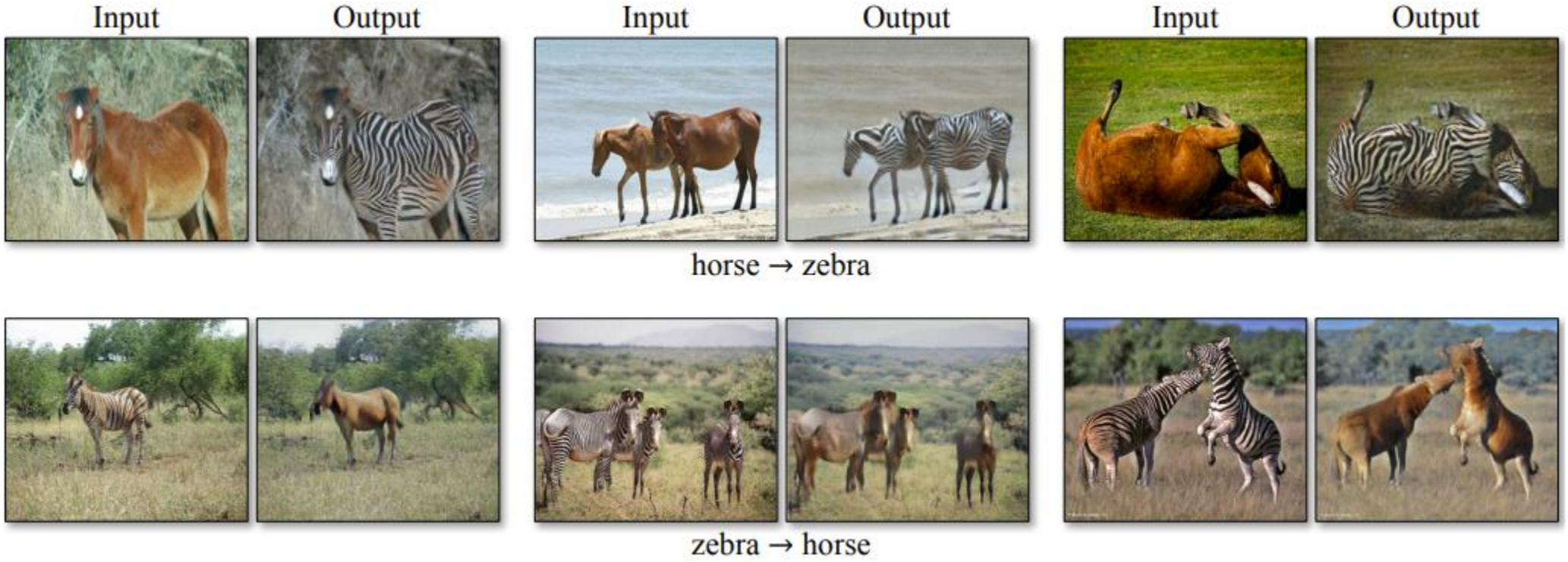
Pix2Pix

- It is relatively easy to construct datasets for color images translation from grayscale images.

- However, it is often difficult to construct datasets for learning in the real world.

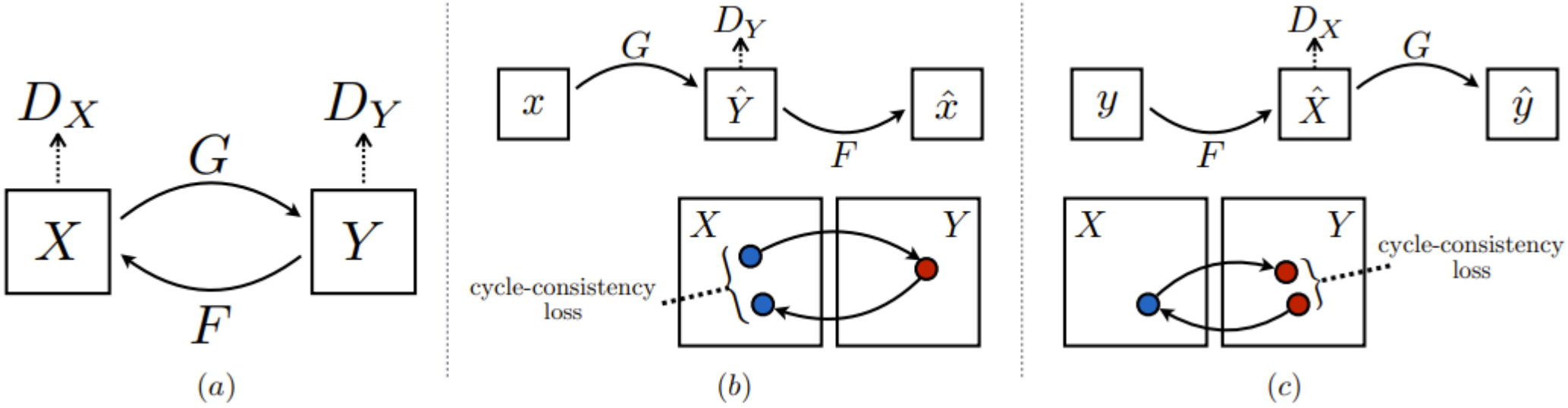
- For example, if you need 1000 images of Monet? :(

3 CycleGAN



Given any two unordered image collections, CycleGAN translates image to image.

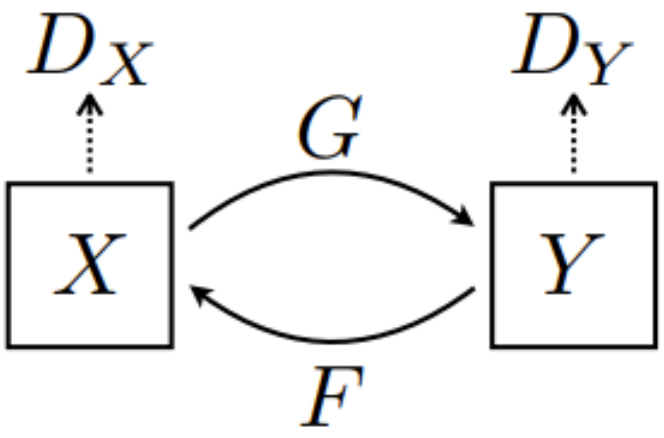
3 CycleGAN



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

CycleGAN contains two mapping functions and associated adversarial discriminator D_Y and D_Y.

3 CycleGAN

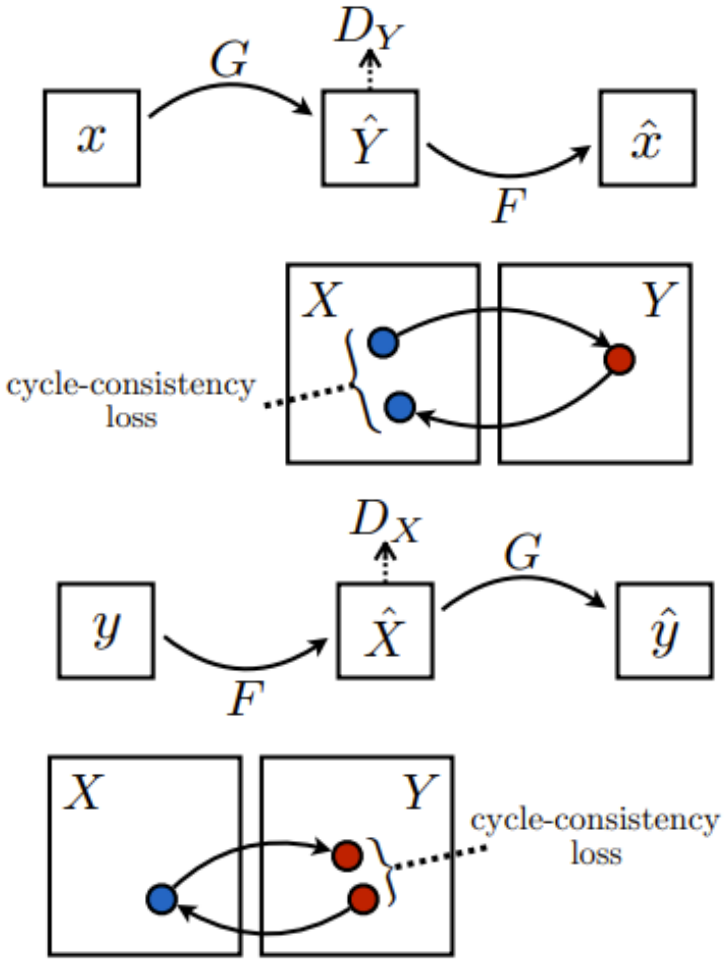


Adversarial loss

$$L_{GAN}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{data}(y)}[logD_Y(y) + \mathbb{E}_{x \sim p_{data}(x)}[logD_Y((G(x))]$$

$$L_{GAN}(F, D_X, X, Y) = \mathbb{E}_{x \sim p_{data}(x)}[logD_X(x) + \mathbb{E}_{y \sim p_{data}(y)}[logD_X((F(y))]$$

3 CycleGAN



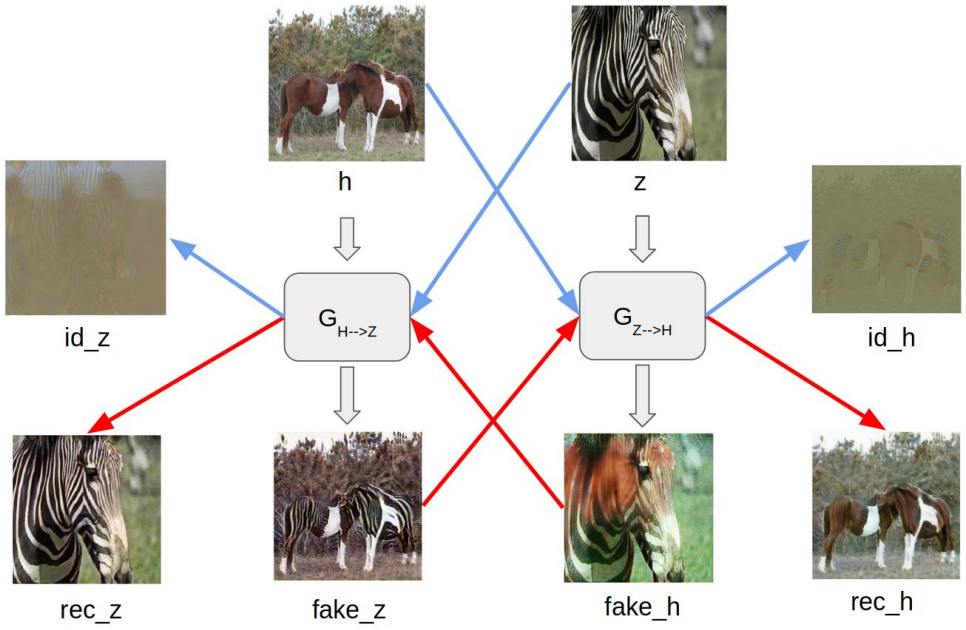
Cycle consistency loss

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

Full objective

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

3 CycleGAN



$G\{H \rightarrow Z\}$ is the generator that transform horse images into zebra images.
$G\{Z \rightarrow H\}$ is the generator that transform zebra images into horse images.

•From h, $G\{H \rightarrow Z\}$ generates $fake\_z$ a fake zebra image. From z, $G\{H \rightarrow Z\}$ generates id_z that should be identical to z.
•From z, $G\{Z \rightarrow H\}$ generates $fake\_h$ a fake horse image. From h, $G\{Z \rightarrow H\}$ generates id_h that should be identical to h.
•From $fake\_z$, $G\{Z \rightarrow H\}$ generates rec_h a reconstructed image of h that should be similar to h.
•From $fake\_h$, $G\{H \rightarrow Z\}$ generates rec_z a reconstructed image of z that should be similar to z.

3 CycleGAN

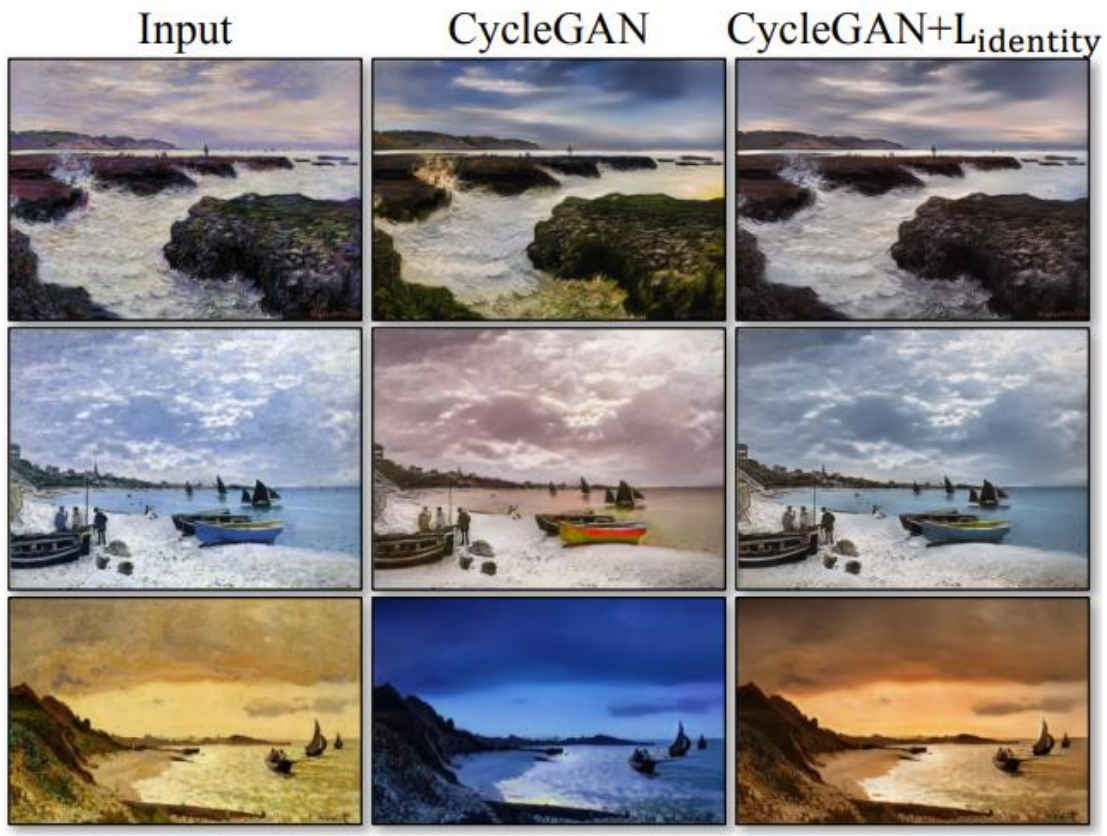
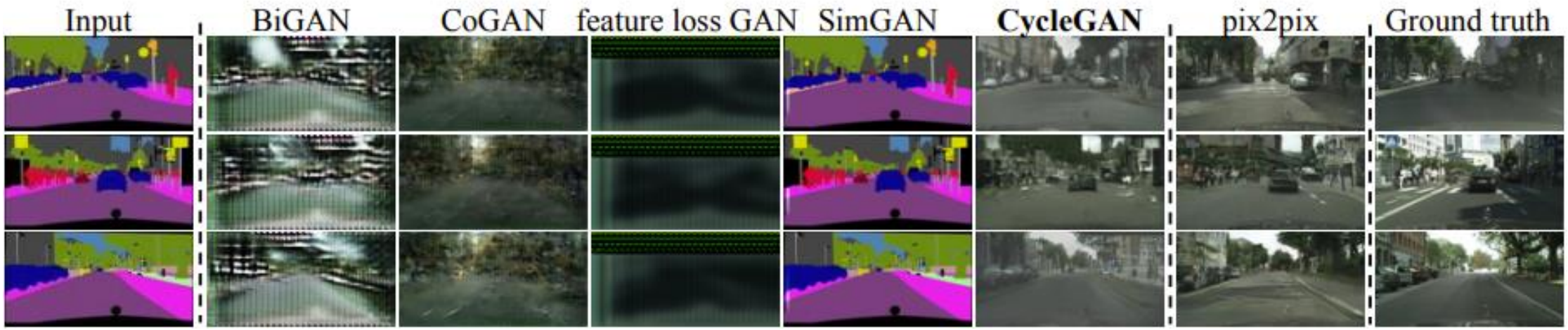


Photo generation from paintings

They found that $L_{identiy}$ is to encourage the mapping to preserve color composition between the input and output, which is helpful for specific tasks.

$$L_{identiy}(G,F) = \mathbb{E}_{y\sim p_{data}(y)}[\|G(y)-y\|_1] + \mathbb{E}_{x\sim p_{data}(x)}[\|F(x)-x\|_1]$$

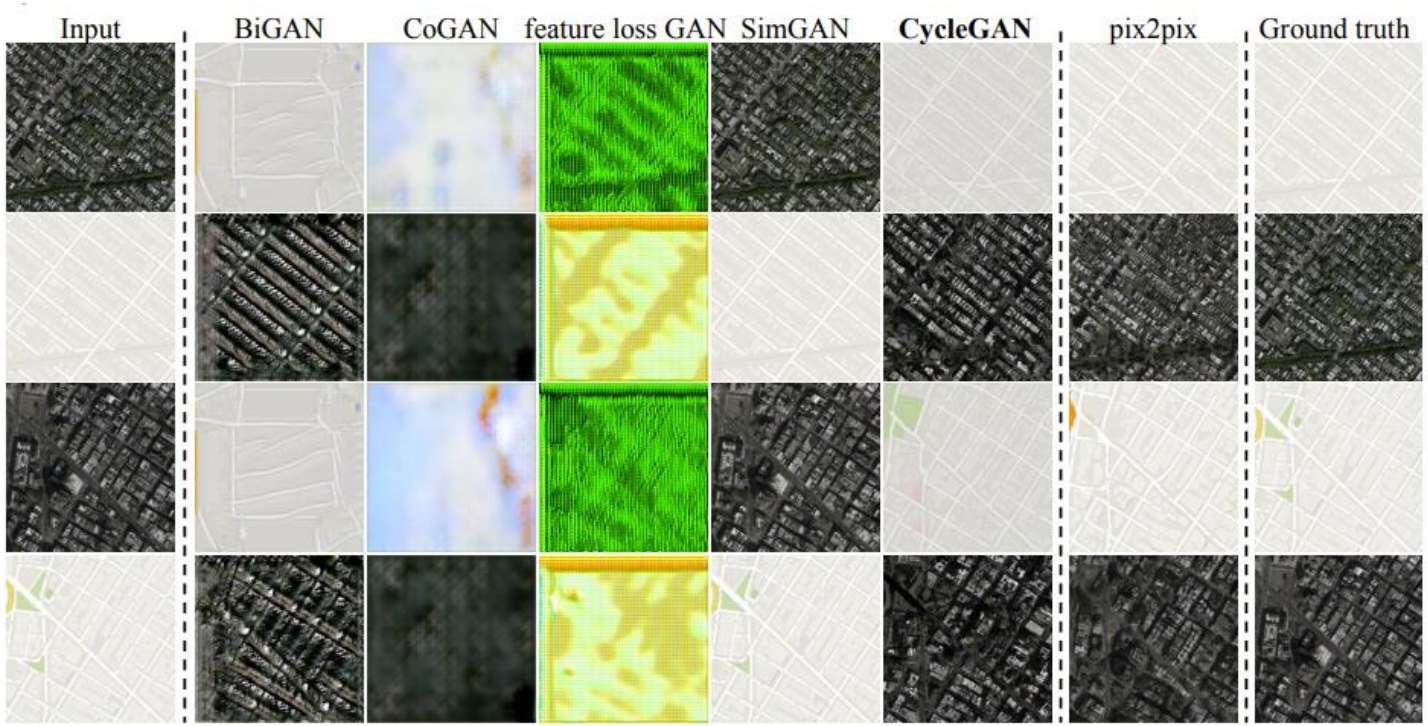
4 Result



Different methods for mapping labels↔photos trained on Cityscapes images.

BiGAN, CoGAN, feature loss + GAN, SimGAN, CycleGAN, pix2pix trained on paired data, and ground truth.

4 Result



Different methods for mapping aerial photos↔maps on Google Maps.

BiGAN, CoGAN, feature loss + GAN, SimGAN, CycleGAN, pix2pix trained on paired data, and ground truth.

4 Result

Loss	Map → Photo % Turkers labeled *real*	Photo → Map % Turkers labeled *real*
CoGAN [32]	0.6% ± 0.5%	0.9% ± 0.5%
BiGAN/ALI [9, 7]	2.1% ± 1.0%	1.9% ± 0.9%
SimGAN [46]	0.7% ± 0.5%	2.6% ± 1.1%
Feature loss + GAN	1.2% ± 0.6%	0.3% ± 0.2%
CycleGAN (ours)	**26.8% ± 2.8%**	**23.2% ± 3.4%**

AMT "real vs fake" test on Google Maps maps↔aerial photos at 256 × 256 resolution.

Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [32]	0.40	0.10	0.06
BiGAN/ALI [9, 7]	0.19	0.06	0.02
SimGAN [46]	0.20	0.10	0.04
Feature loss + GAN	0.06	0.04	0.01
CycleGAN (ours)	**0.52**	**0.17**	**0.11**
pix2pix [22]	0.71	0.25	0.18

FCN-scores for different methods, evaluated on Cityscapes labels→photo.

Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [32]	0.45	0.11	0.08
BiGAN/ALI [9, 7]	0.41	0.13	0.07
SimGAN [46]	0.47	0.11	0.07
Feature loss + GAN	0.50	0.10	0.06
CycleGAN (ours)	**0.58**	**0.22**	**0.16**
pix2pix [22]	0.85	0.40	0.32

Classification performance of photo→labels for different methods on cityscapes.

4 Result

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	**0.55**	**0.18**	**0.12**
GAN + backward cycle	0.39	0.14	0.06
CycleGAN (ours)	0.52	0.17	0.11

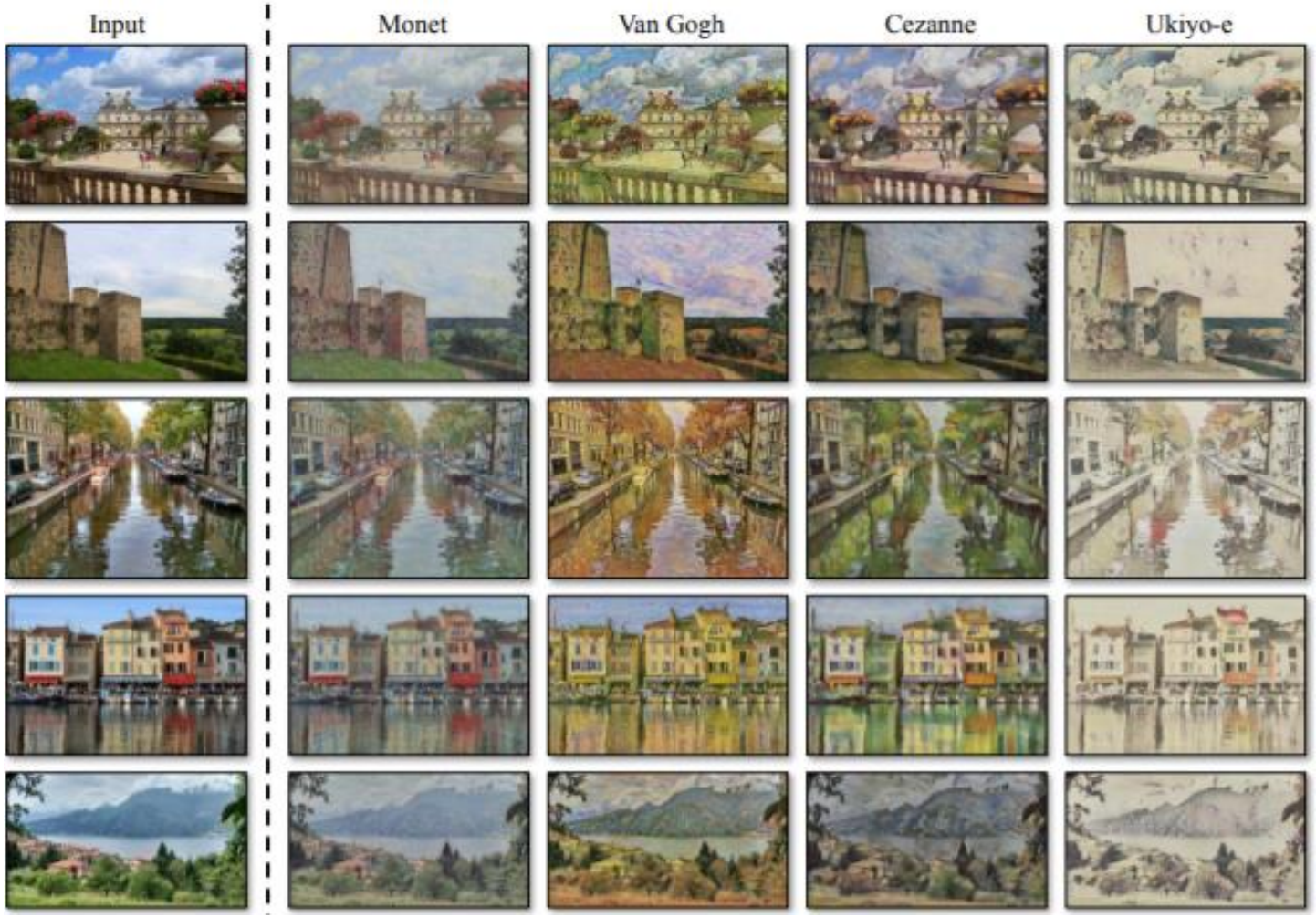
FCN-scores for different variants of our method, evaluated on Cityscapes labels→photo.

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)	**0.58**	**0.22**	**0.16**

Classification performance of photo→labels for different losses, evaluated on Cityscapes.

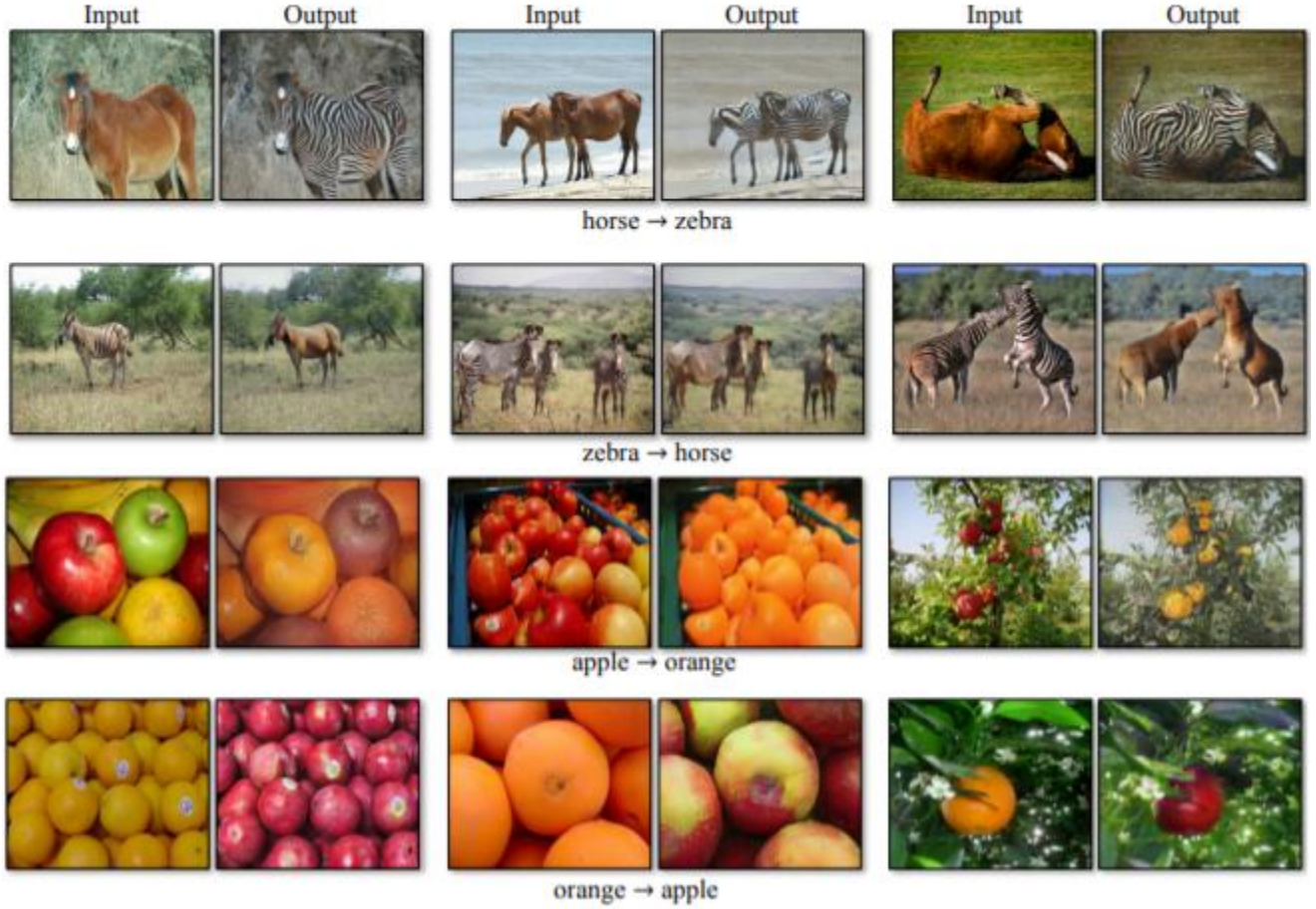
4 Result

Collection style transfer



Result

Object transfiguration

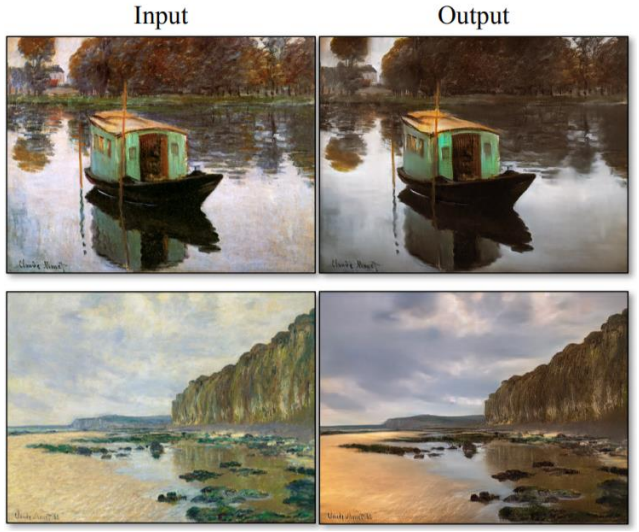


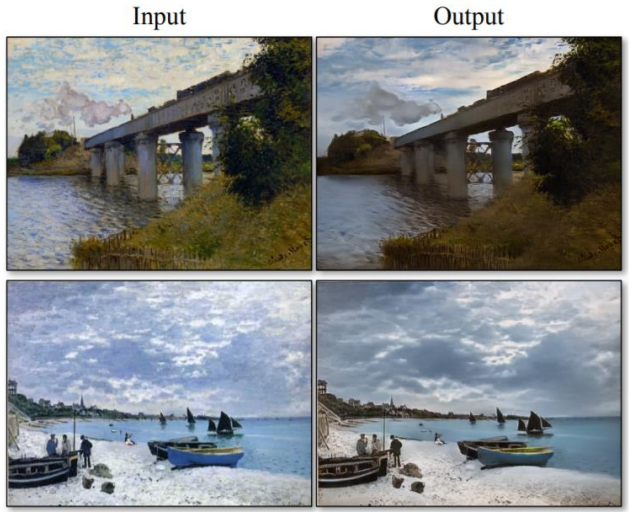
4 Result

Season transfer

4 Result

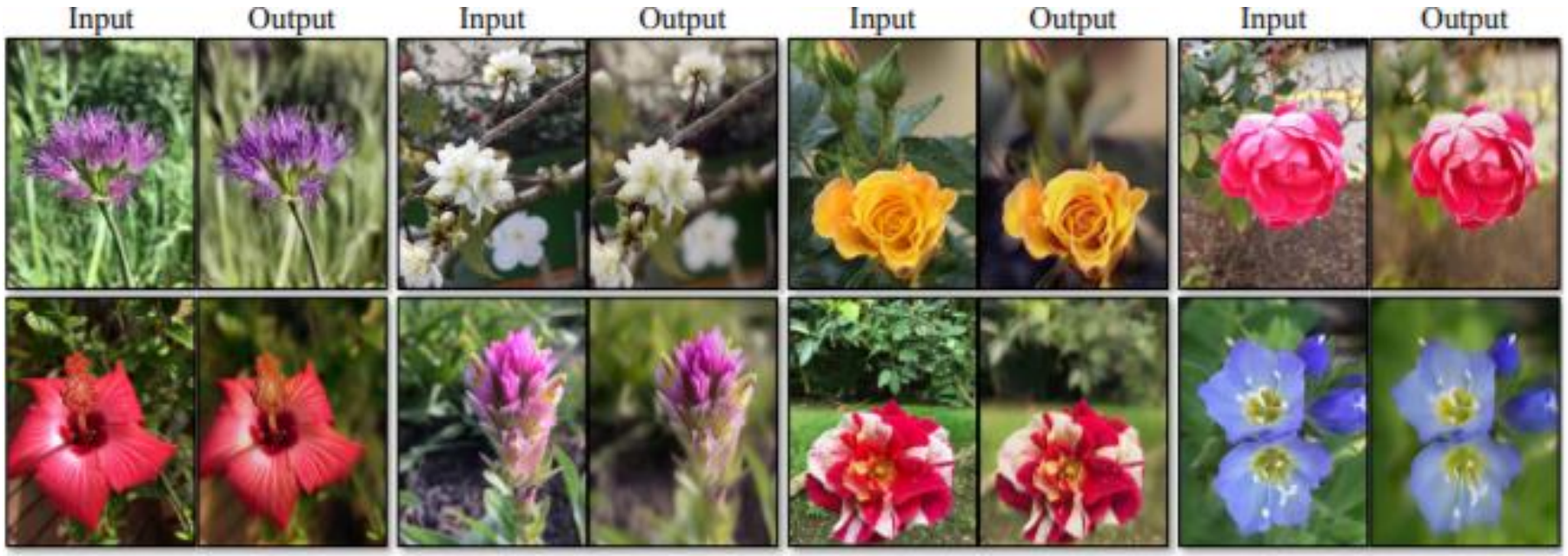
Photo generation from paintings





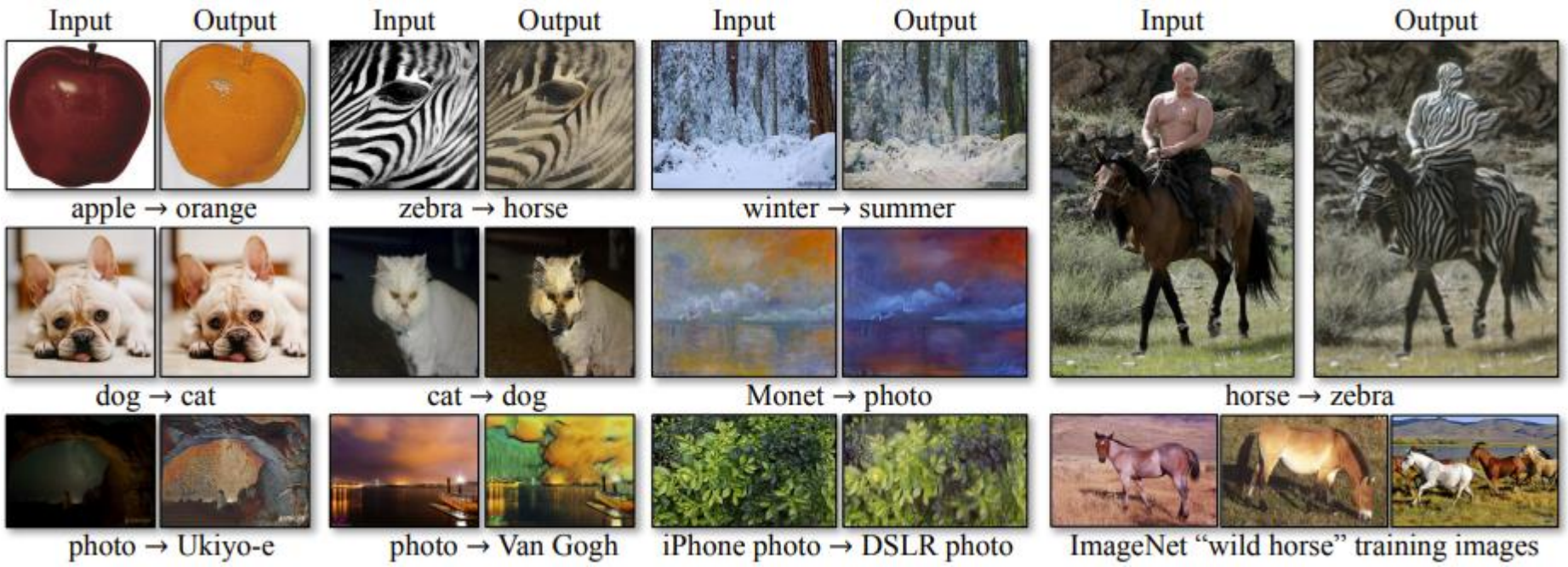
Result

Photo enhancement



Result

Typical failure cases of CycleGAN



Reference

- https://arxiv.org/pdf/1703.10593.pdf
- https://arxiv.org/pdf/1611.07004.pdf
- http://www.kwangsiklee.com/2018/03/cyclegan%EC%9D%B4-%EB%AC%B4%EC%97%87%EC%9D%B8%EC%A7%80-%EC%95%8C%EC%95%84%EB%B3%B4%EC%9E%90/
- https://towardsdatascience.com/overview-of-cyclegan-architecture-and-training-afee31612a2f
- https://taeoh-kim.github.io/blog/image2image/