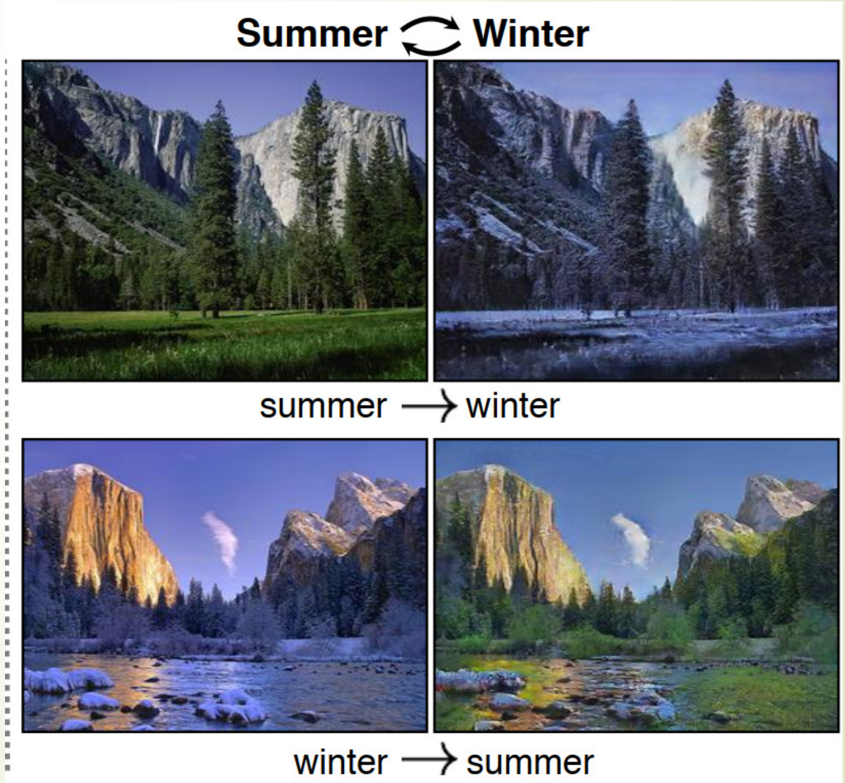
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Network

Jun-Yan Zhu
Taesung Park
Phillip Isola
Alexei A. Efros

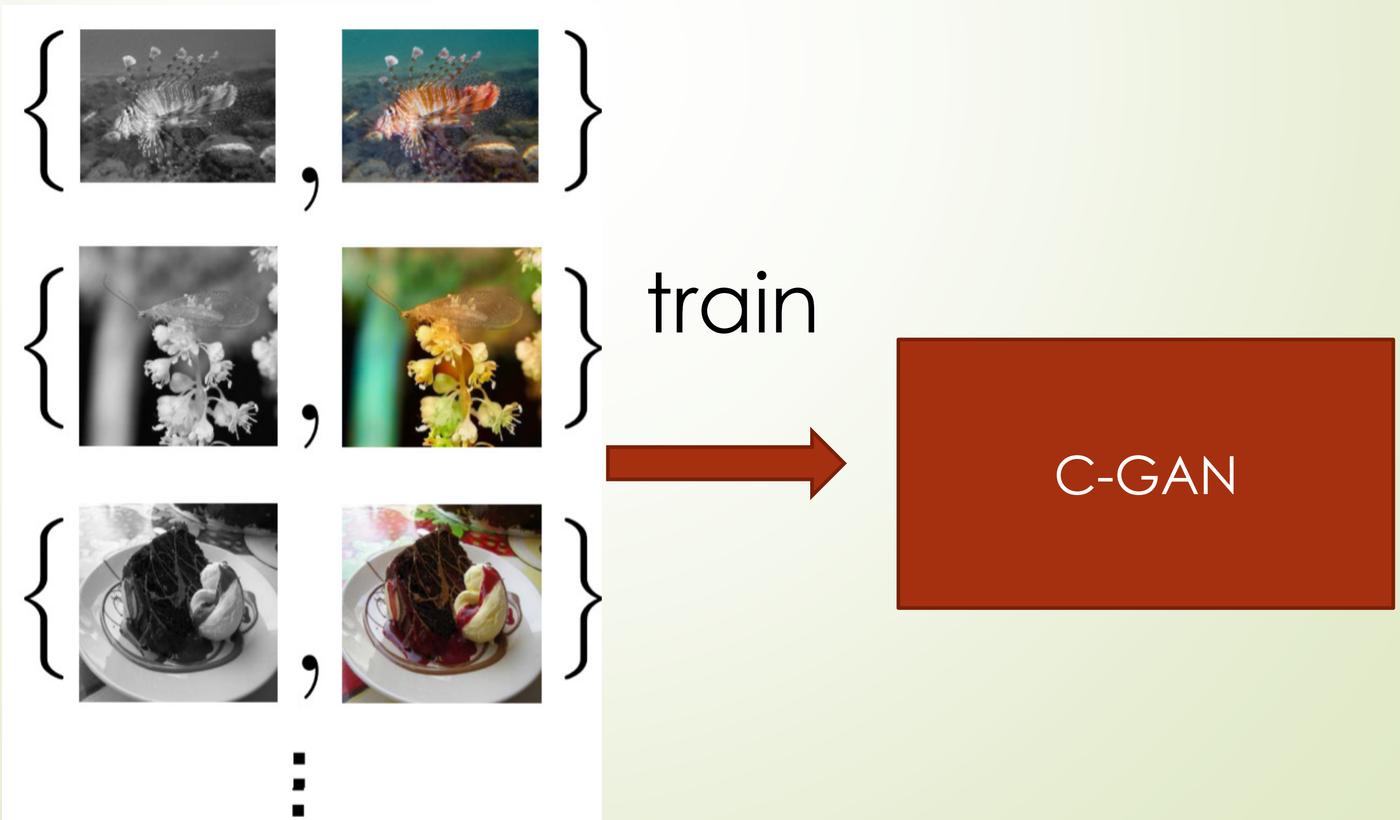
2017



Presented by: Waseem Wishahi – Nov 2021

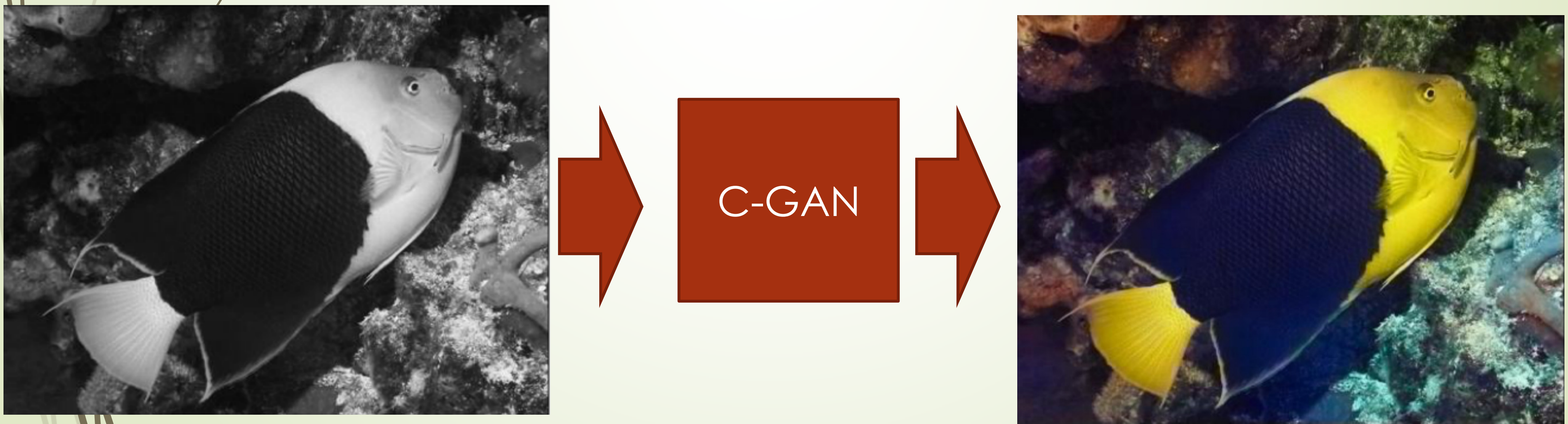
Previously…

- We learned about Image to Image translation
- **Get paired images from different domains**
- Use Conditional GAN to learn the mapping between the two



Previously…

- We learned about Image to Image translation
- **Get paired images from different domains**
- Use Conditional GAN to learn the mapping between the two



Paired Image to Image

- However, for many tasks, paired training data will not be available!

Landscape photos ↔ Van Gogh Paintings

How can we even obtain such data?!

Unpaired Image to Image Translation

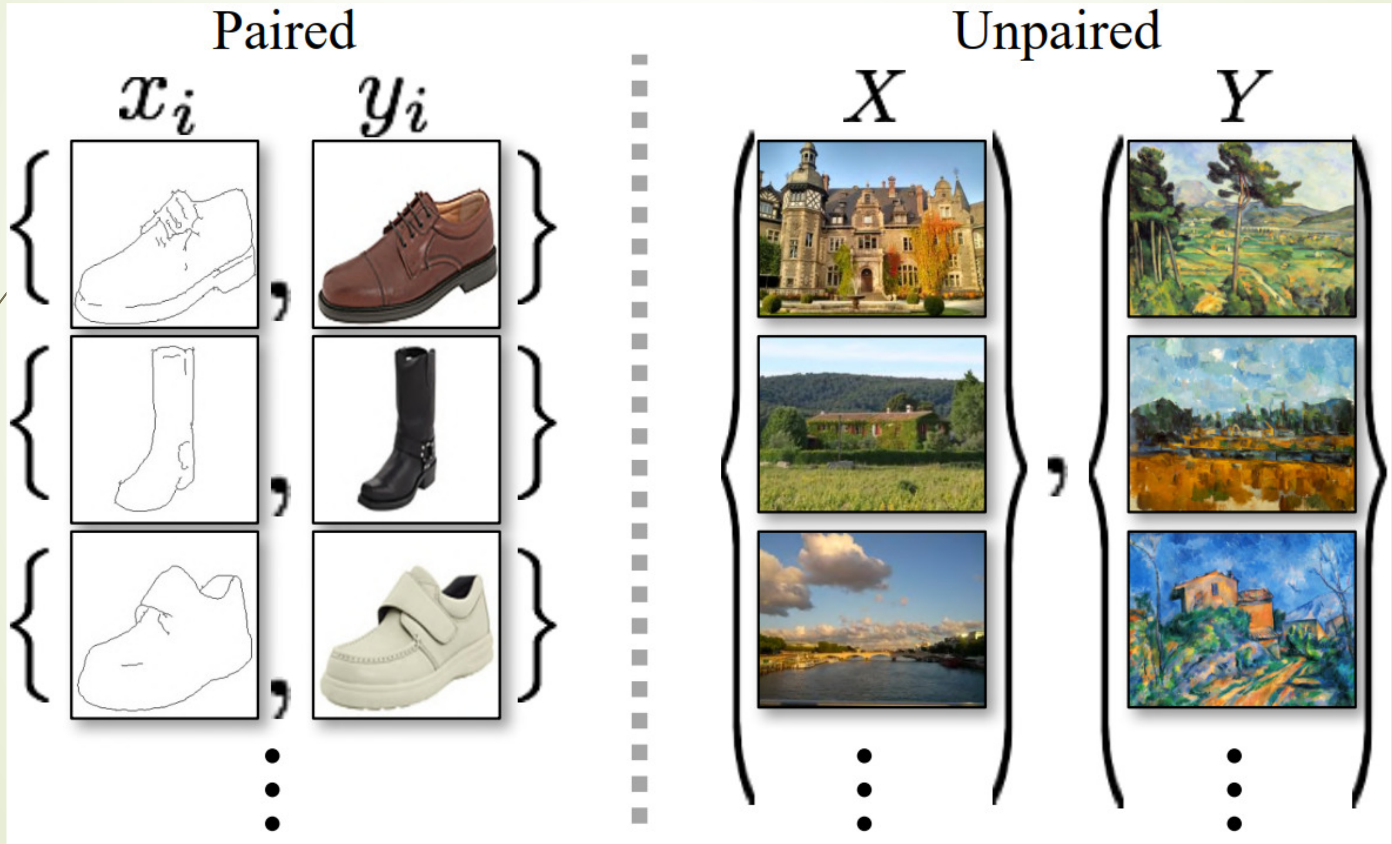
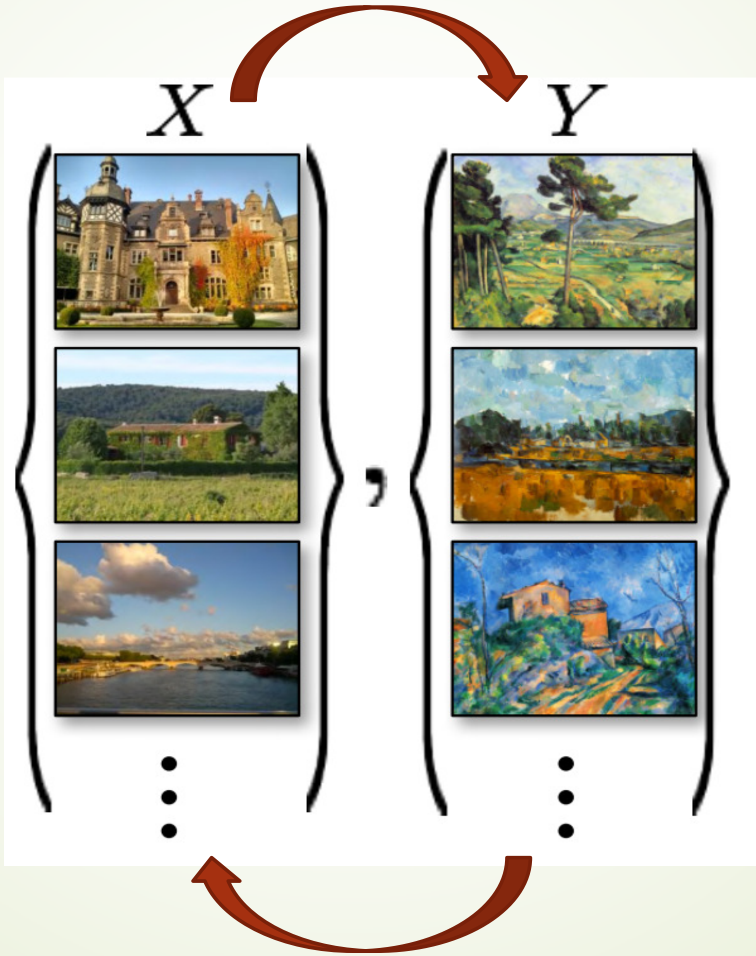


Image to Image

$G: X \to Y$

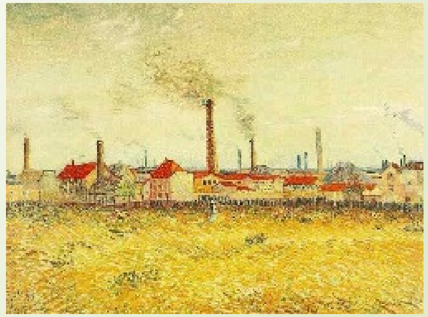


$F: Y \to X$

Constraints

- We want to preserve the distribution:
- $\hat{y} = G(x)$ indistinguishable from $y \in Y$

Van Gogh Domain



x

$G(x)$

Constraints

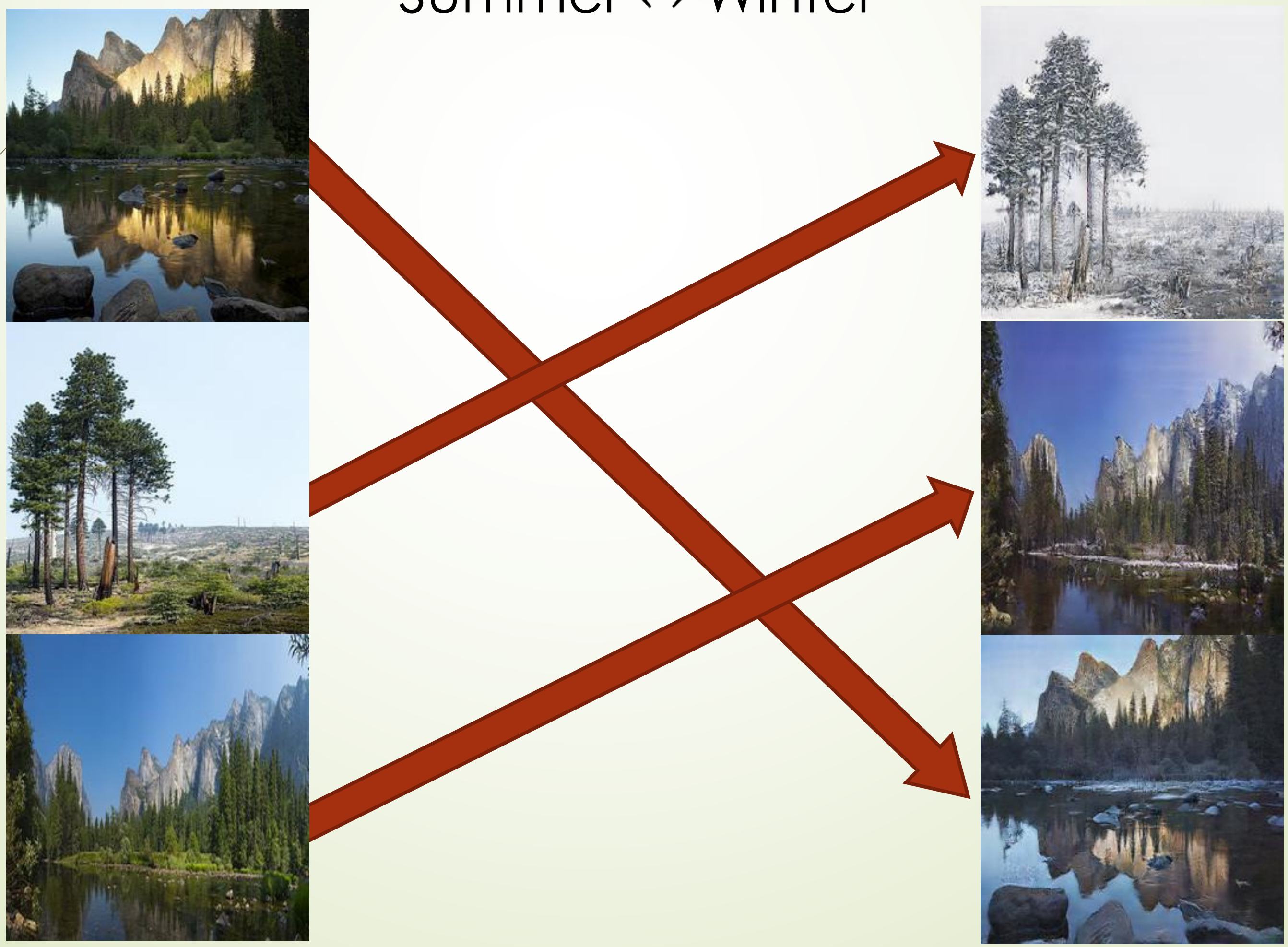
- If we only force distribution, we could get something like this:

Summer ↔ Winter

Constraints

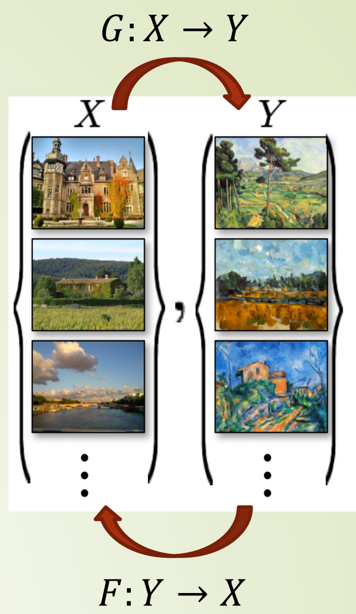
- Need to also preserve input-output connection

Summer ↔ Winter



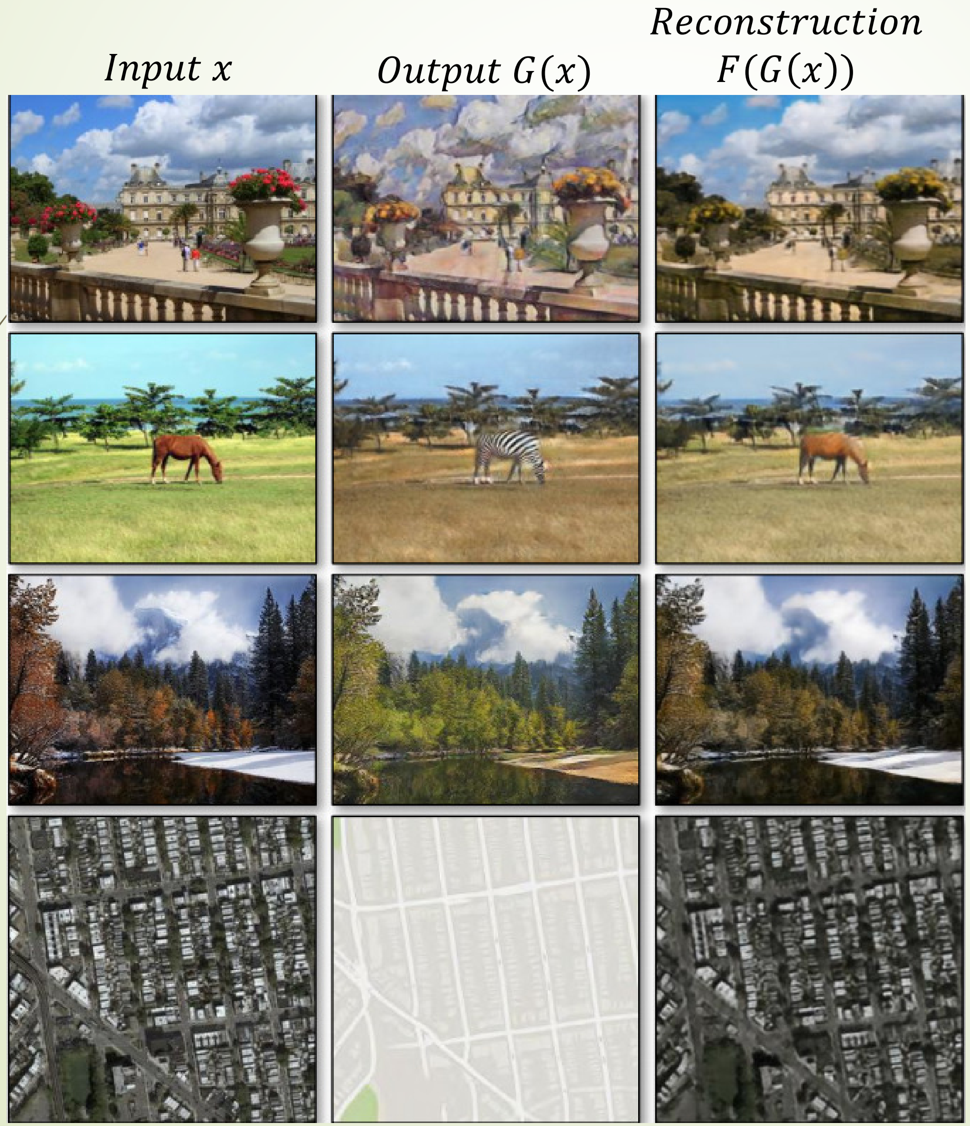
Constraints

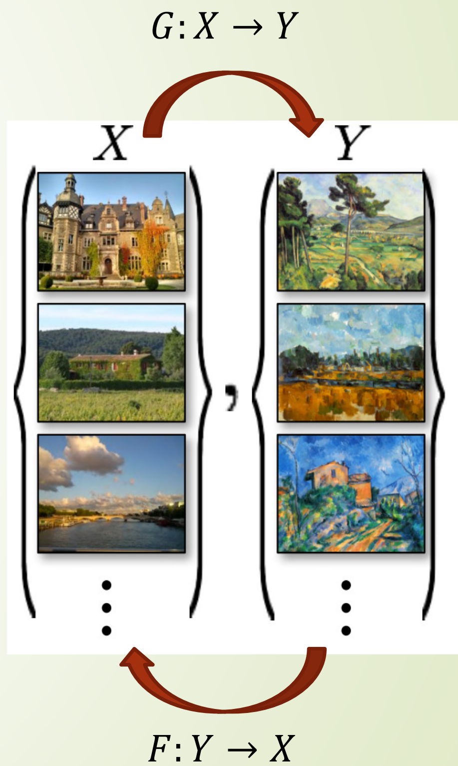
- Cycle Consistency



Constraints

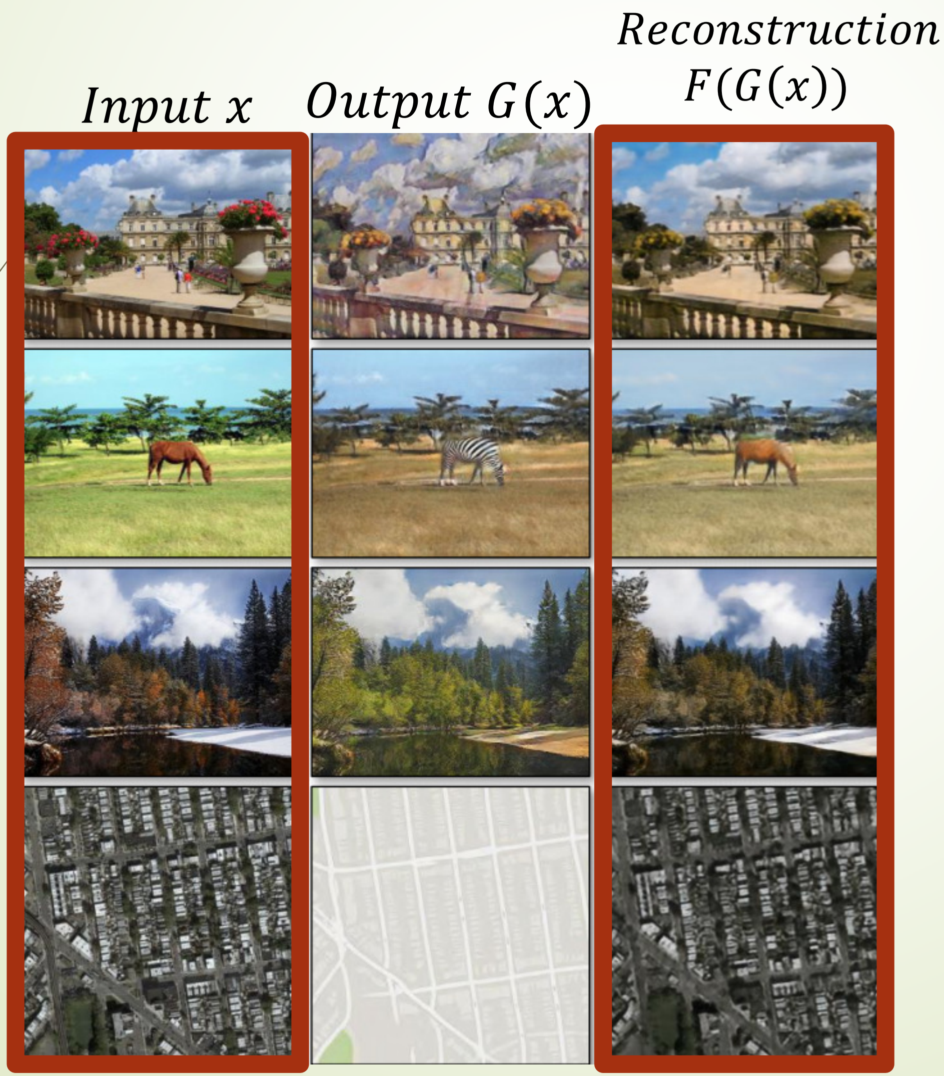
- Cycle Consistency





Constraints

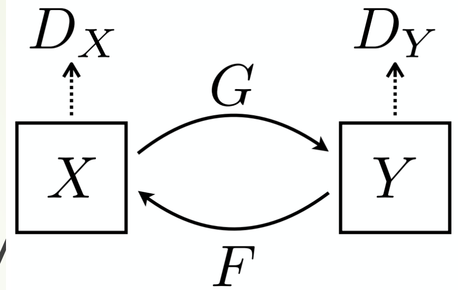
- Cycle Consistency

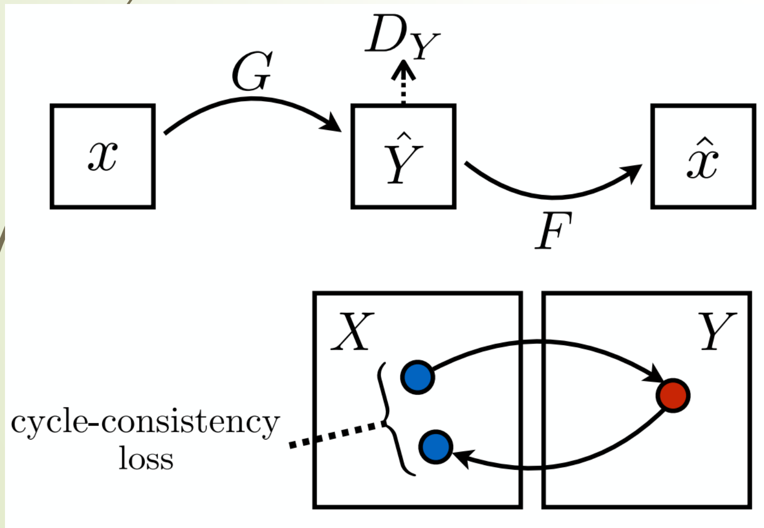


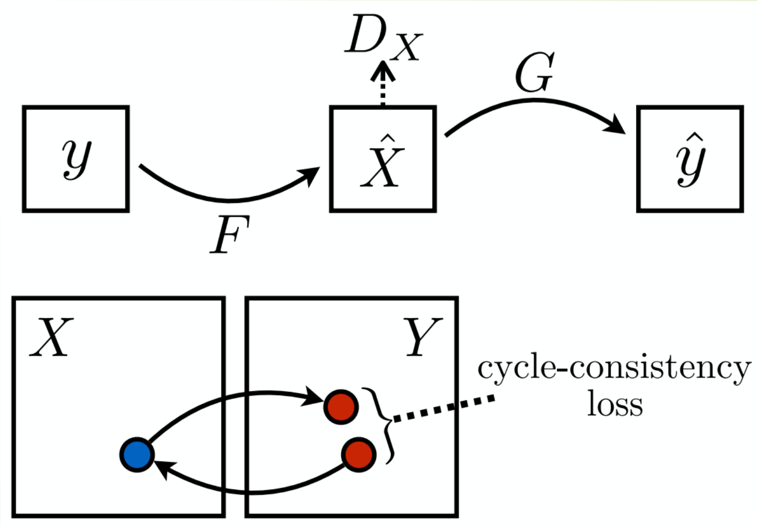
Need to minimize the difference

Constraints

- Preserve Distribution
- Preserve Cycle-Consistency







Loss function(s)

For preserving distribution $G: X \rightarrow Y$:

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

$$\min_{G} \max_{D_Y} \mathfrak{L}_{GAN}(G, D_Y, X, Y)$$

Loss function(s)

For preserving distribution $F: Y \rightarrow X$:

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

$$\min_F \max_{D_X} \mathfrak{L}_{GAN}(F, D_X, Y, X)$$

Loss function(s)

- For Preserving Cycle-Consistency:

  Objective: $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ $\qquad y \rightarrow F(y) \rightarrow G(F(y)) \approx y$

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

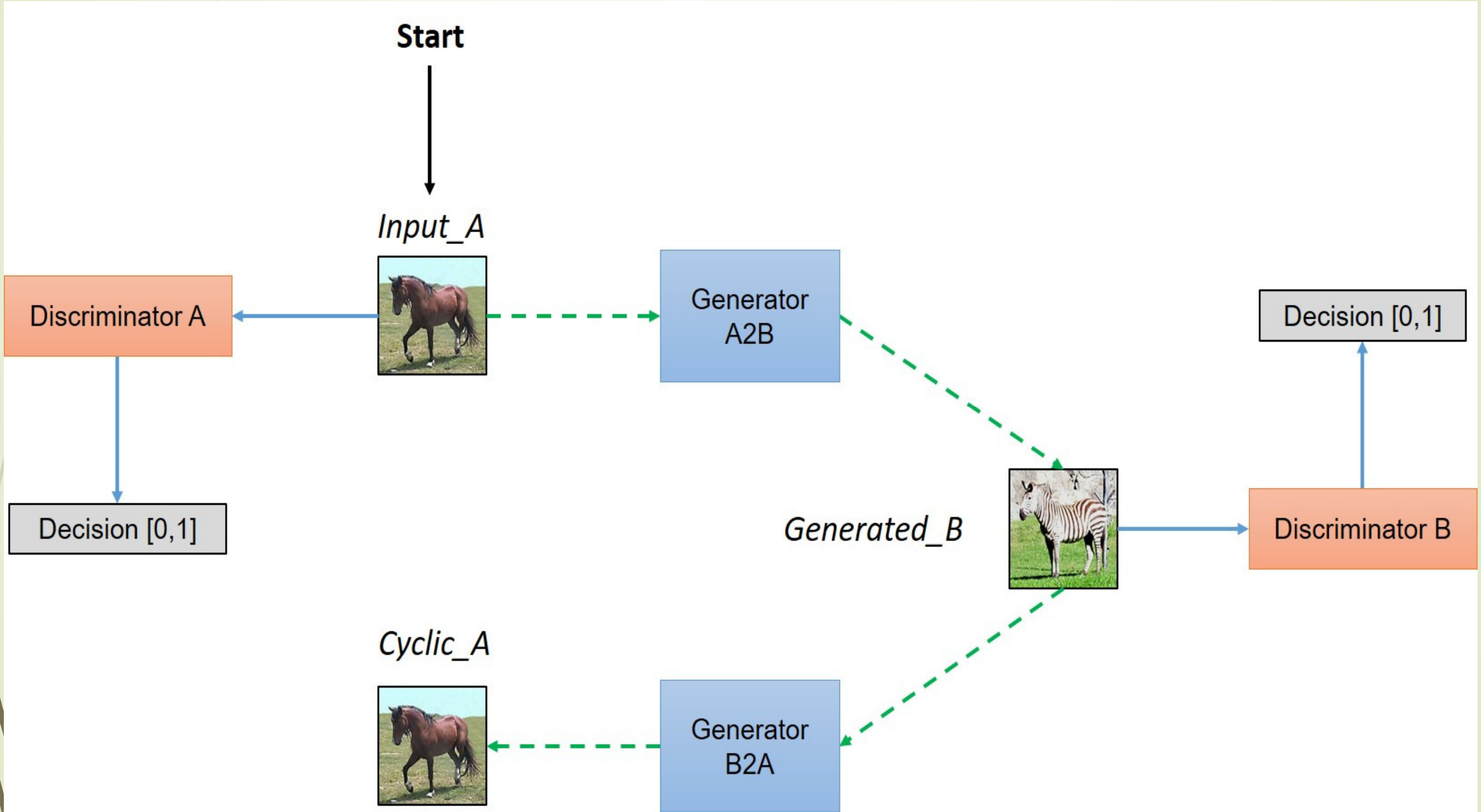
Full objective

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

Hyper parameter for relative importance

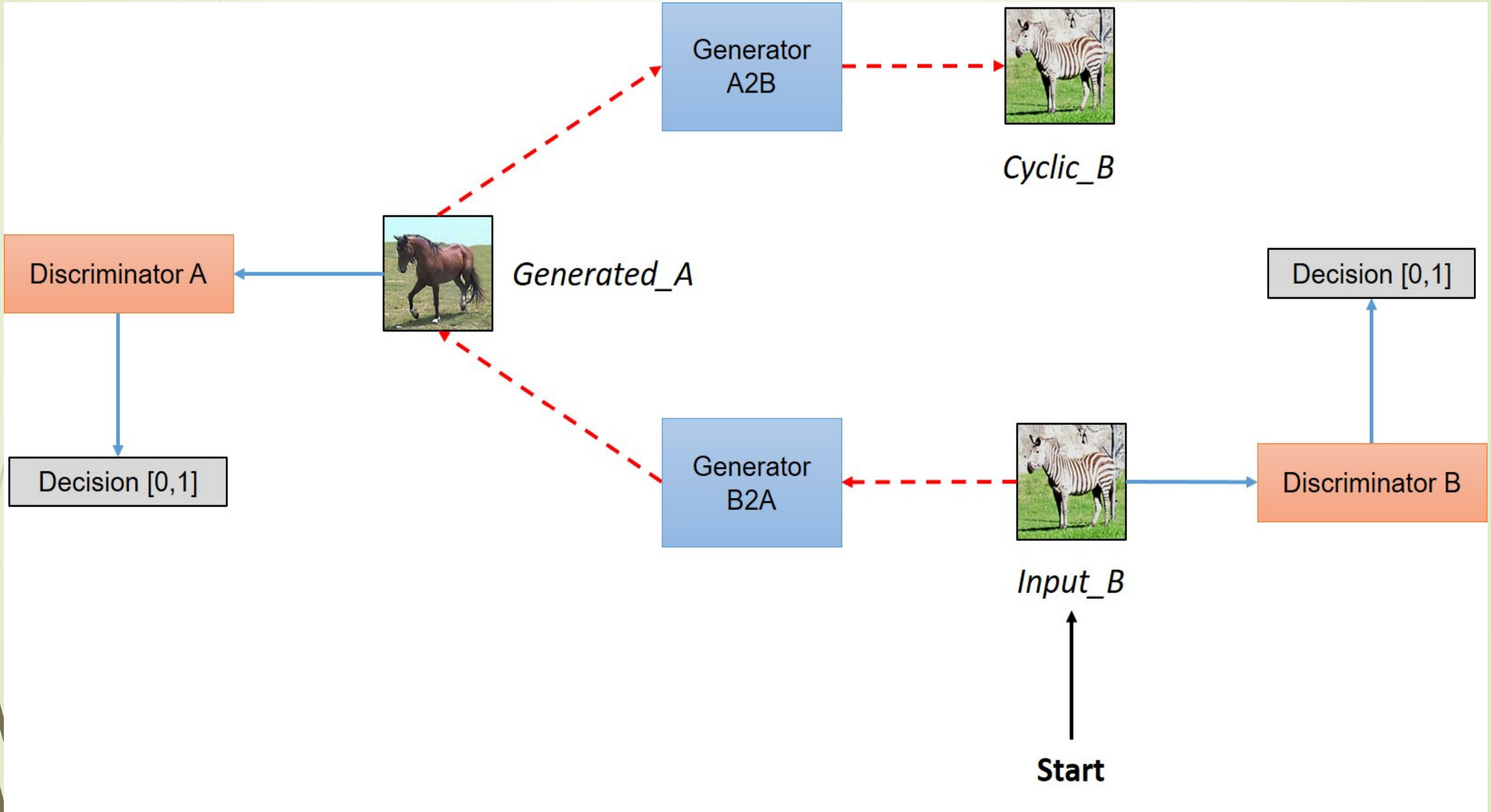
$$\boxed{G^*,F^* = \min_{F,G}\max_{D_X,D_Y}\mathfrak{L}(G,F,D_X,D_Y)}$$

Network architecture (1/2)



hardikbansal.github.io/CycleGANBlog

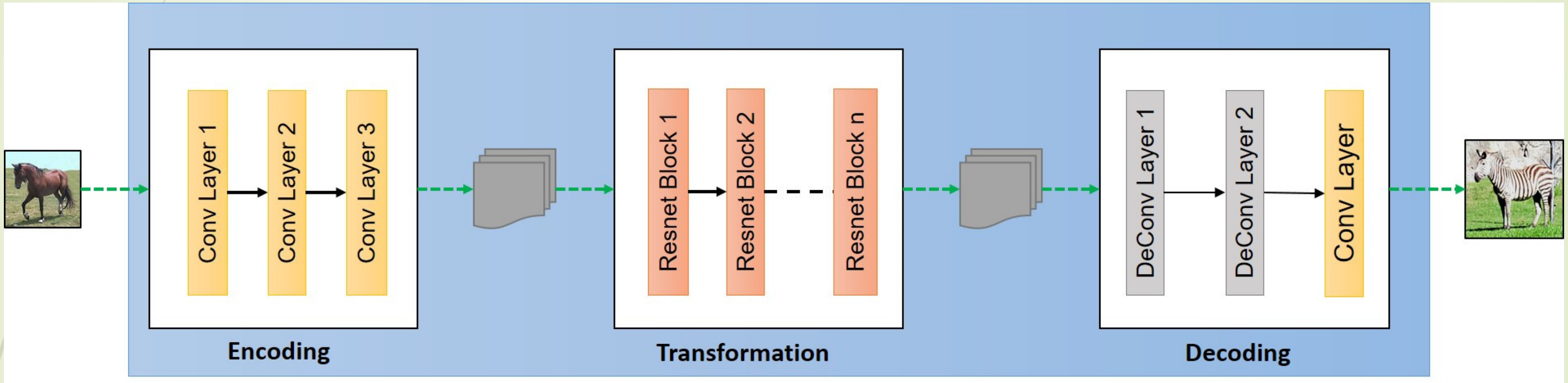
Network architecture (2/2)



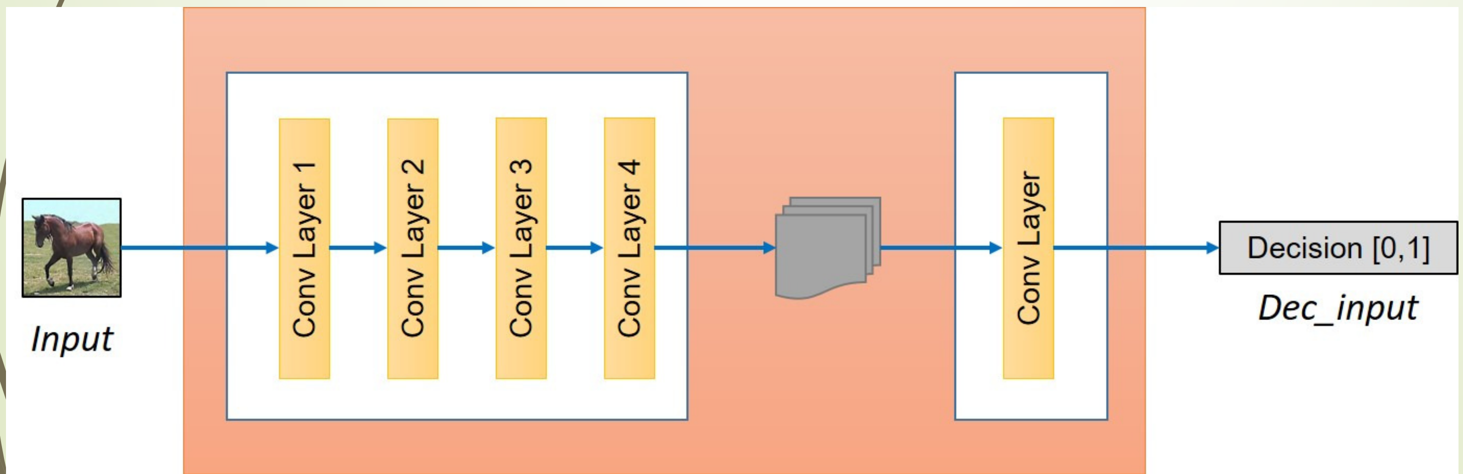
hardikbansal.github.io/CycleGANBlog

Network architecture

High level view of Generators

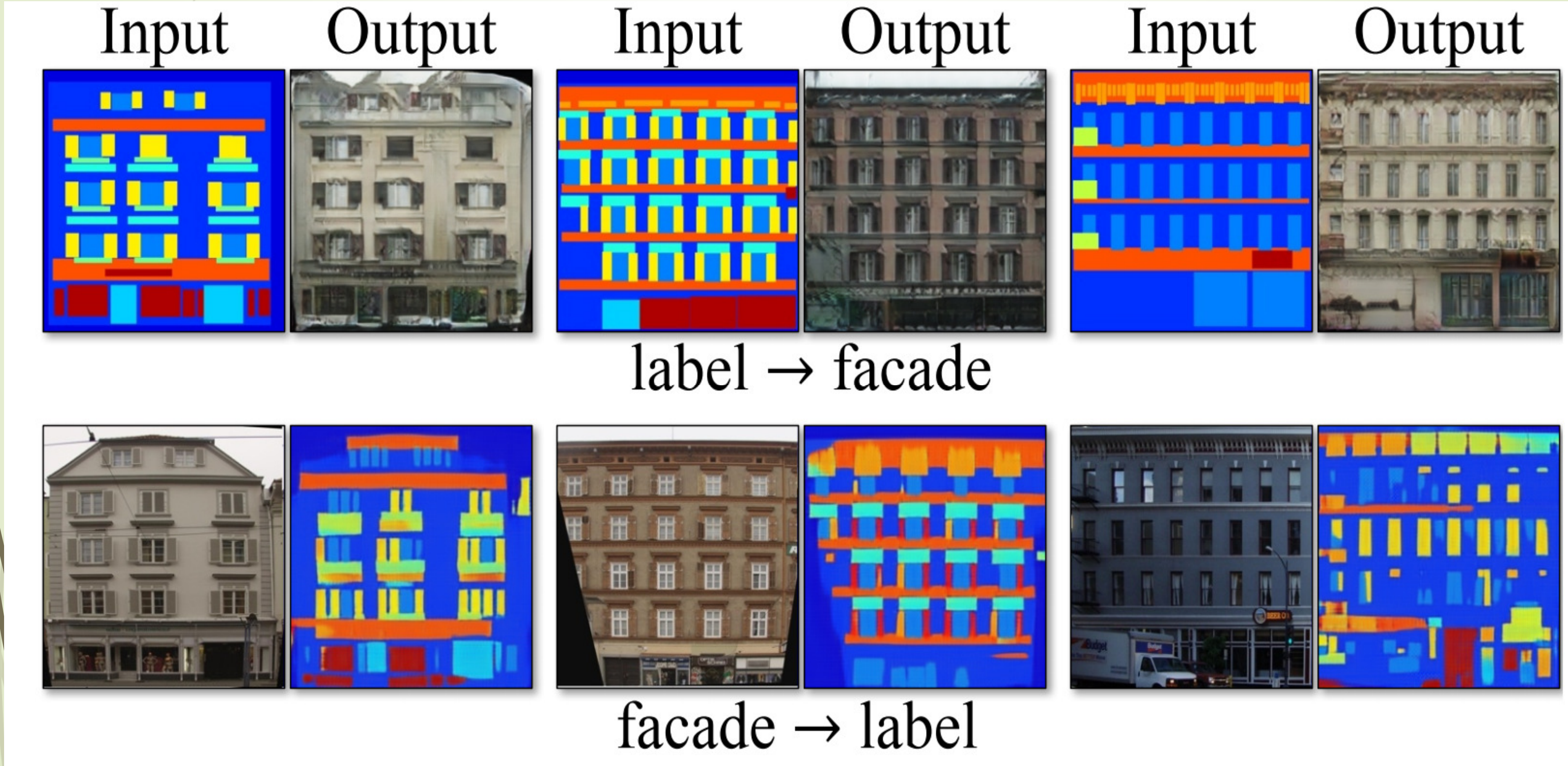


High level view of Discriminator

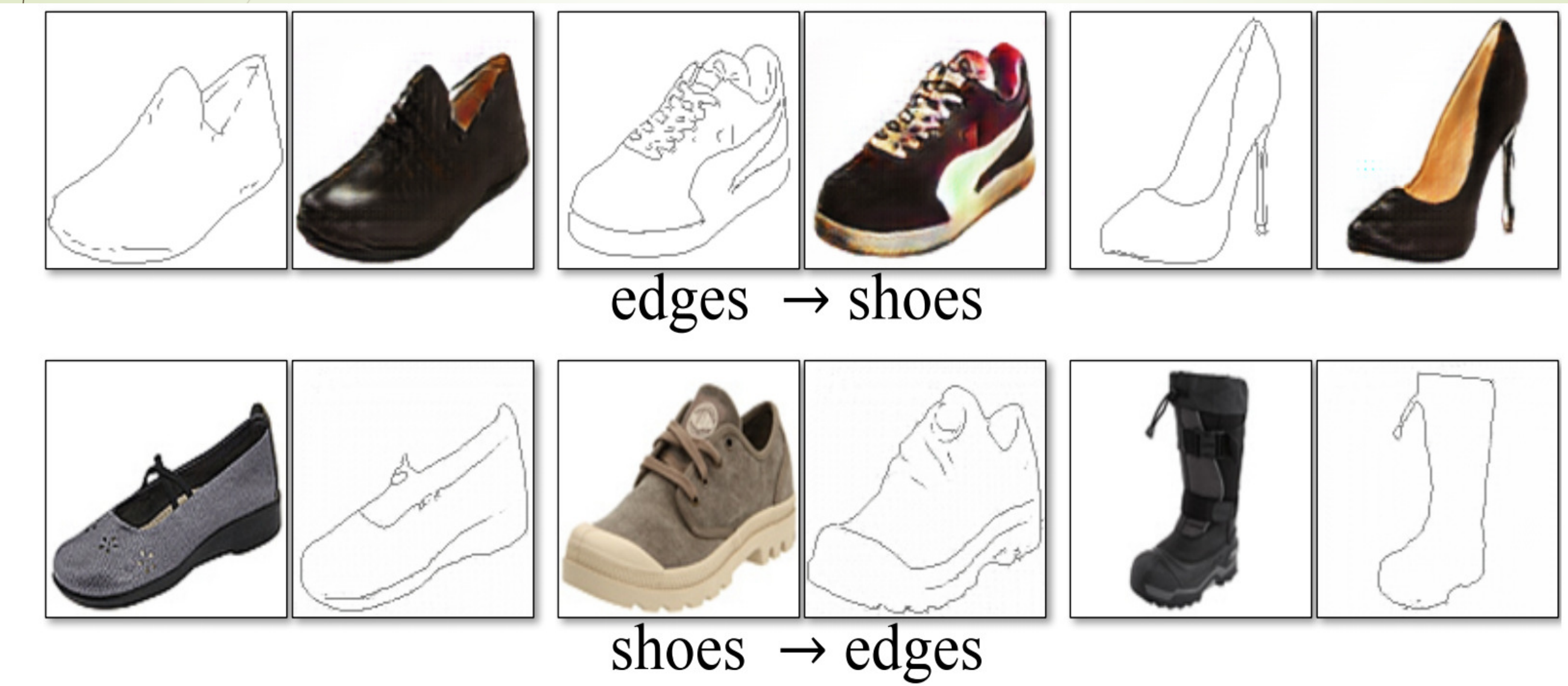


hardikbansal.github.io/CycleGANBlog

Results


Input
Output
Input
Output
Input
Output
label → facade
facade → label

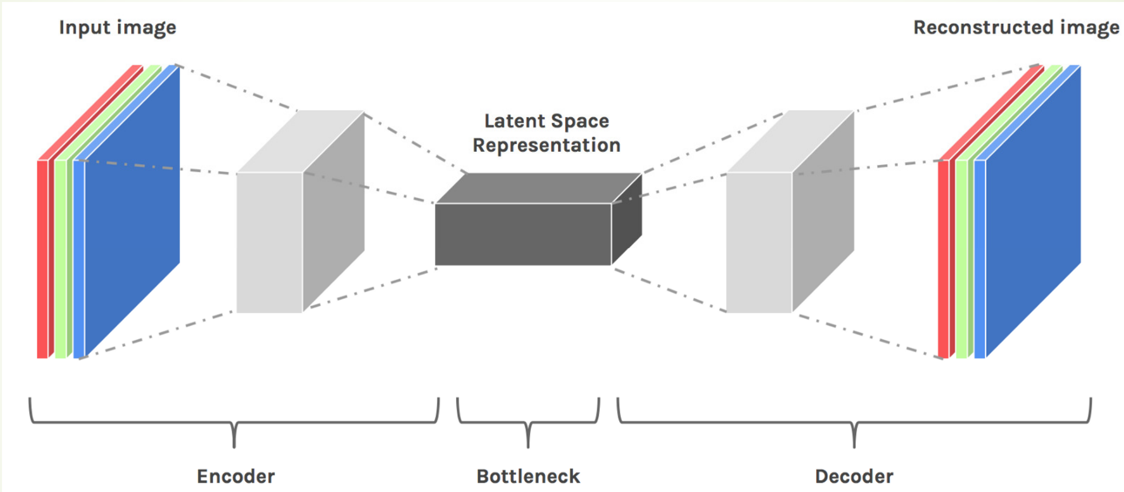
Results



Different approaches

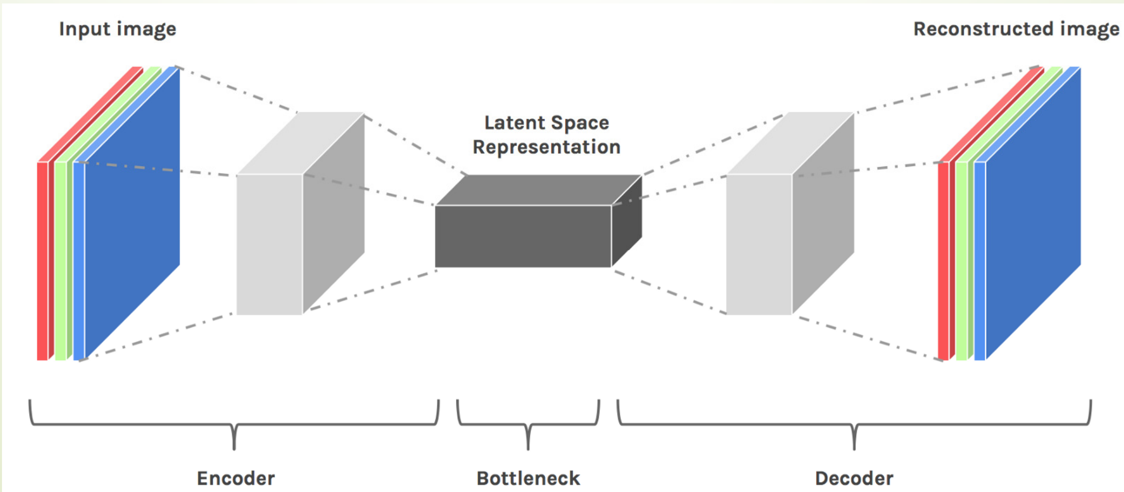
- CoGAN:

Z (noise)



X (domain)

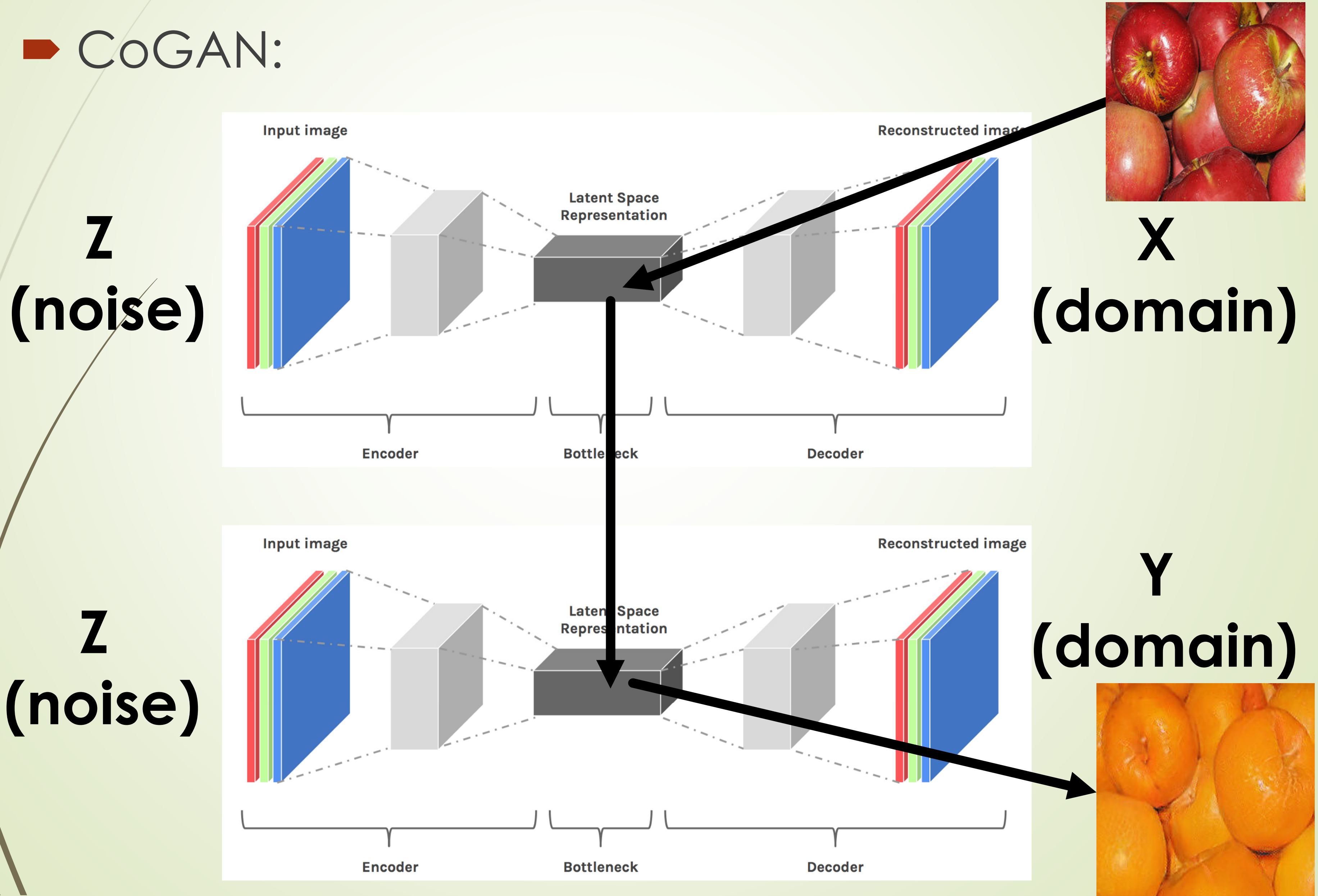
Z (noise)



Y (domain)

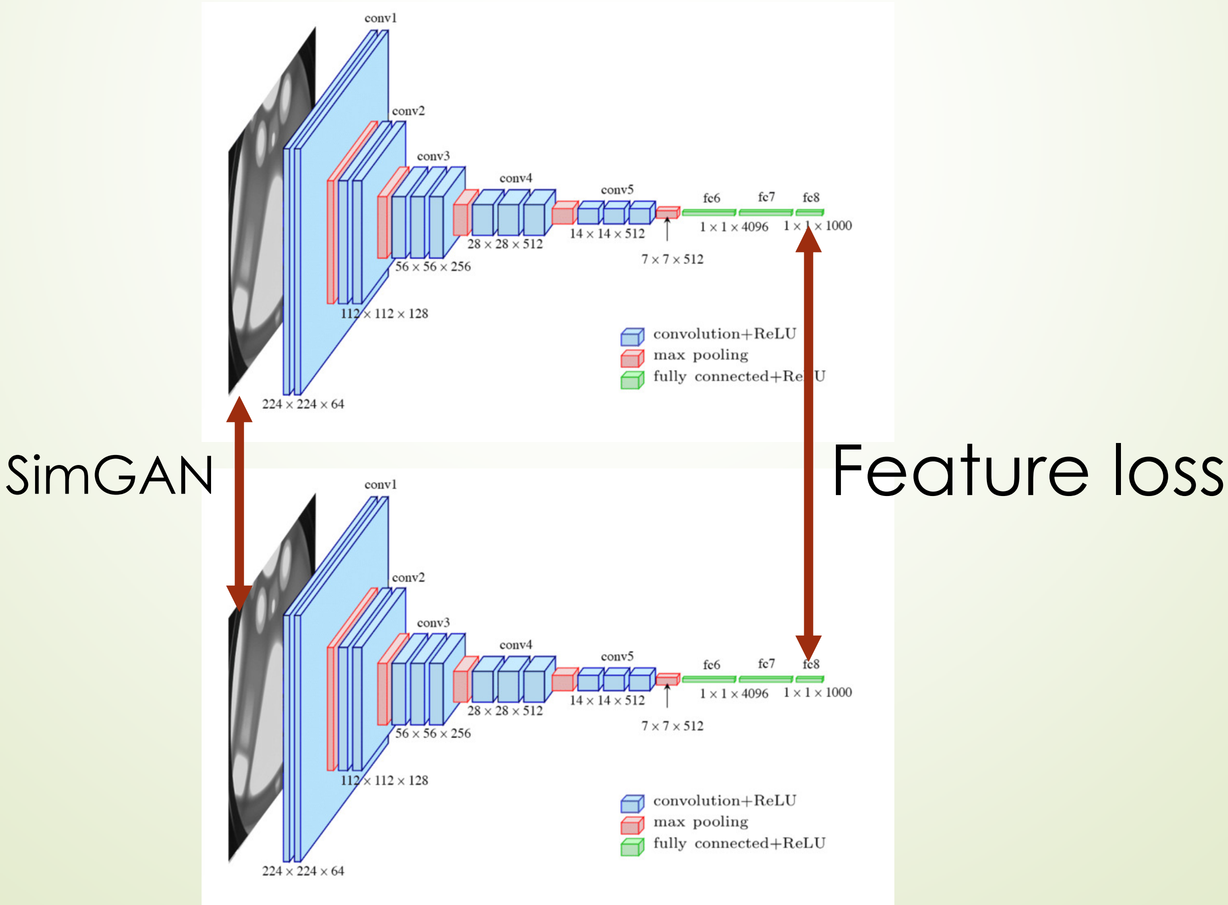
Different approaches

- CoGAN:



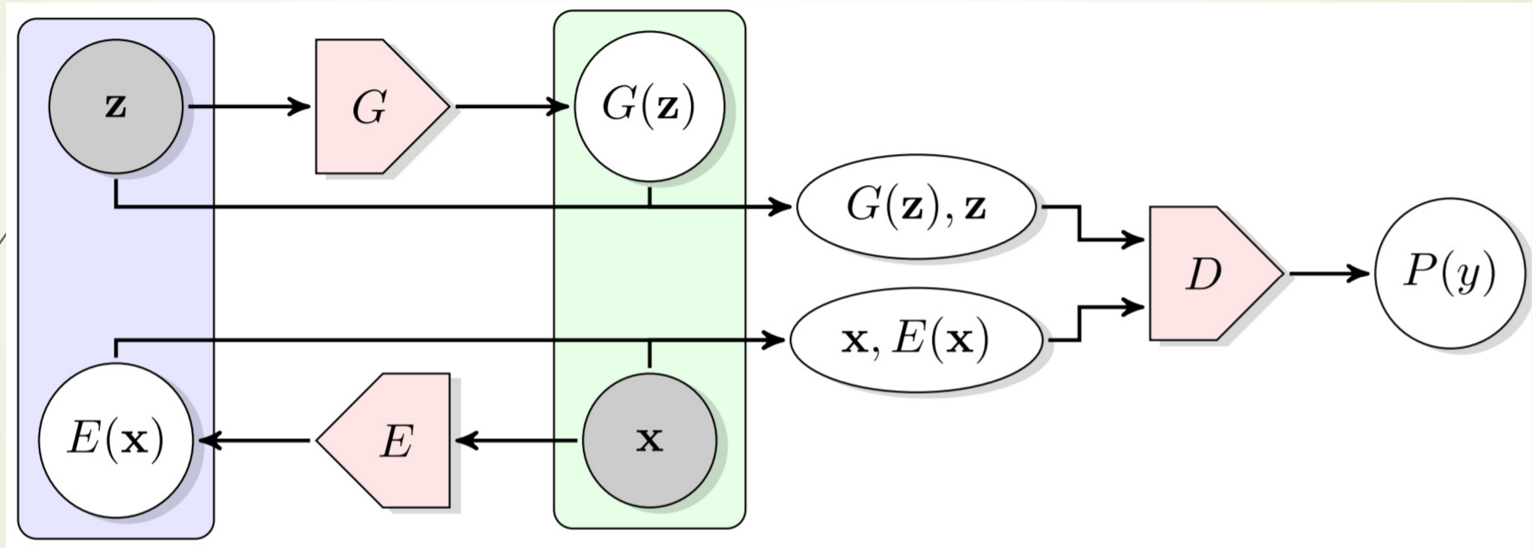
Different approaches

- SimGAN: Uses GAN with L_1 loss $||x - G(x)||_1$
- SimGAN + Feature loss: instead of L_1 on RGB, compute loss on convoluted images



Different approaches

- BiGAN: Learn Mapping $G: X \to Y$, then try to find its inverse.



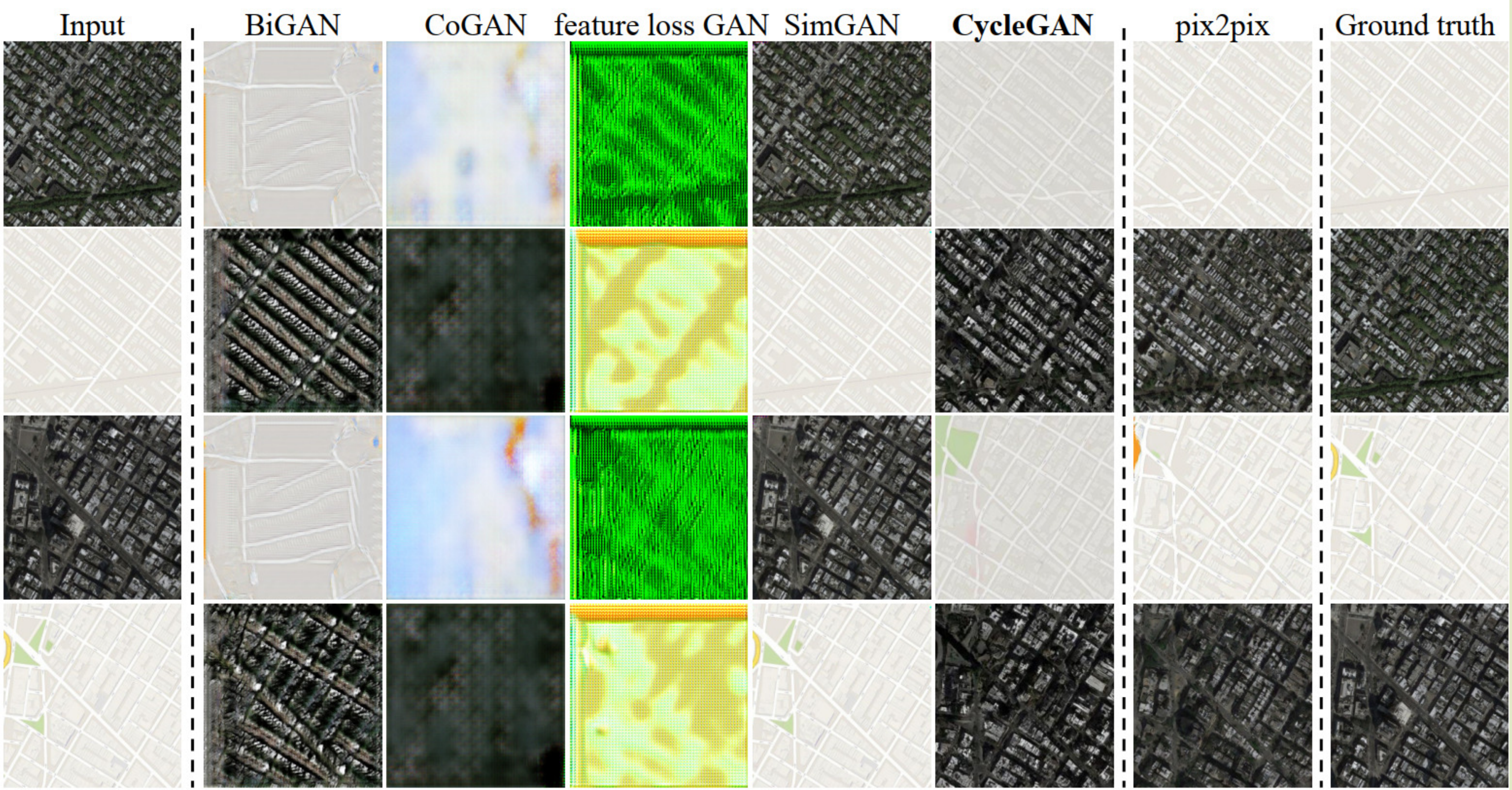
- pix2pix: C-GAN, trained on paired dataset, acts as an upper bound

Evaluation

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 maps-aerial photos at 256 × 256 resolution.

Results compared to different approaches



FCN scores on result vs ground truth

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

$labels \rightarrow photo$

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

$photo \rightarrow labels$

Recap on loss functions

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

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

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

Ablation study

CycleGAN loss: $\boxed{\mathfrak{L}_{GAN}(G, D_y, X, Y) + \mathfrak{L}_{GAN}(G, D_X, Y, X) + \lambda \mathfrak{L}_{cyc}(G, F)}$

GAN alone: $\boxed{\mathfrak{L}_{GAN}(G, D_y, X, Y) + \mathfrak{L}_{GAN}(G, D_X, Y, X)}$

Cycle alone: $\boxed{\mathfrak{L}_{cyc}(G, F)}$

Ablation study

GAN+forward:

$$\mathfrak{L}_{GAN}\left(G, D_y, X, Y\right) + \mathfrak{L}_{GAN}(G, D_X, Y, X) + \lambda\mathfrak{L}'_{cyc}(G, F)$$

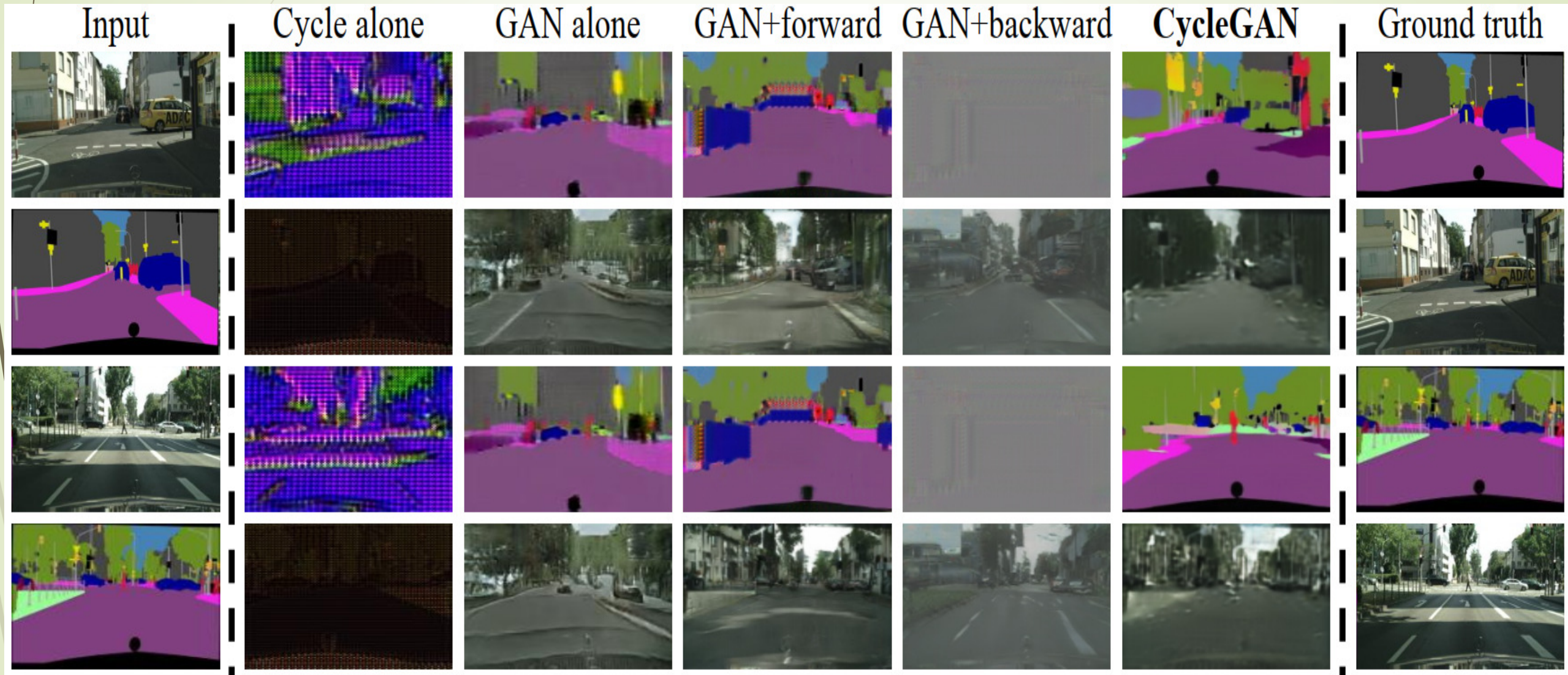
$$\mathfrak{L}'_{cyc}(G, F) = \mathbb{E}_{x\sim p_{data}(x)}\left[||F\left(G(x)\right) - x||_1\right]$$

GAN+backward:

$$\mathfrak{L}_{GAN}\left(G, D_y, X, Y\right) + \mathfrak{L}_{GAN}(G, D_X, Y, X) + \lambda\mathfrak{L}''_{cyc}(G, F)$$

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

Ablation study



$$forward\ cycle: F(G(x)) \approx x \qquad backward\ cycle: G(F(y)) \approx y$$

Ablation study

$labels \rightarrow photo$

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

$photo \rightarrow labels$

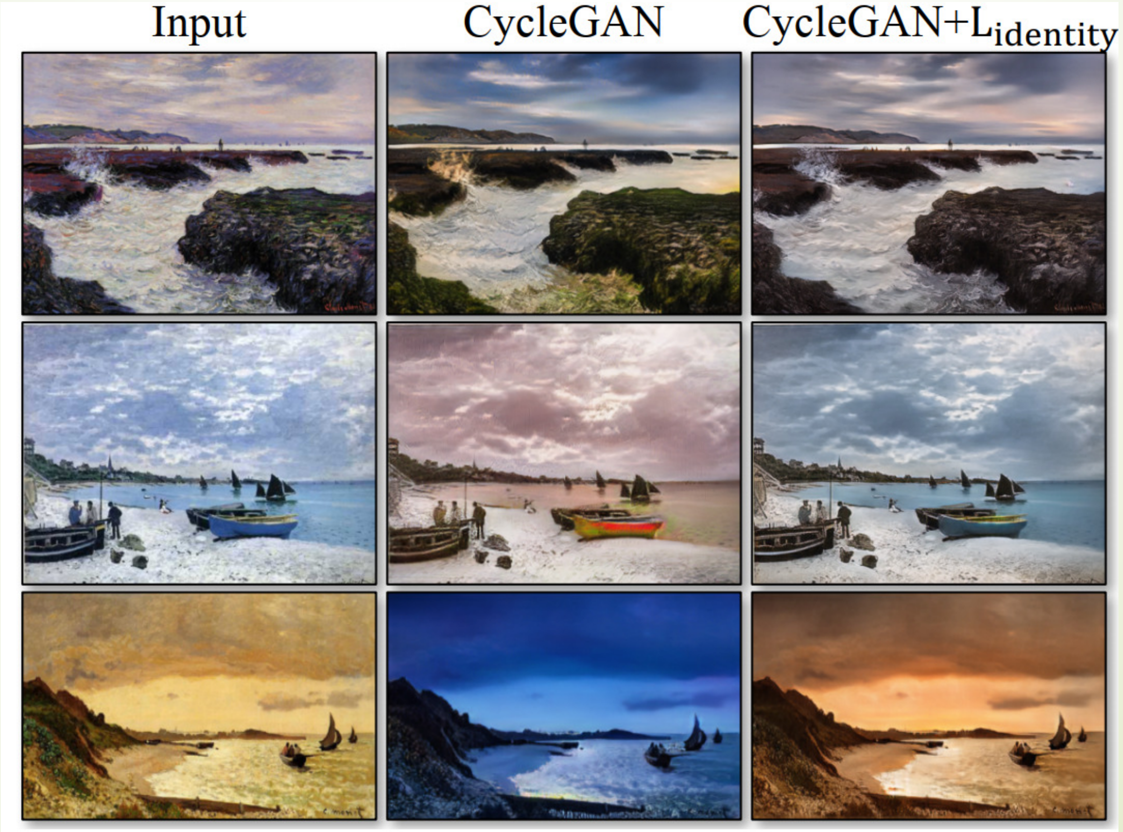
Loss	Per-pixel acc.	Per-class acc.	Class IOU
Cycle alone	0.10	0.05	0.02
GAN alone	0.53	0.11	0.07
GAN + forward cycle	0.49	0.11	0.07
GAN + backward cycle	0.01	0.06	0.01
CycleGAN (ours)	**0.58**	**0.22**	**0.16**

Notable applications

Photo generation from paintings

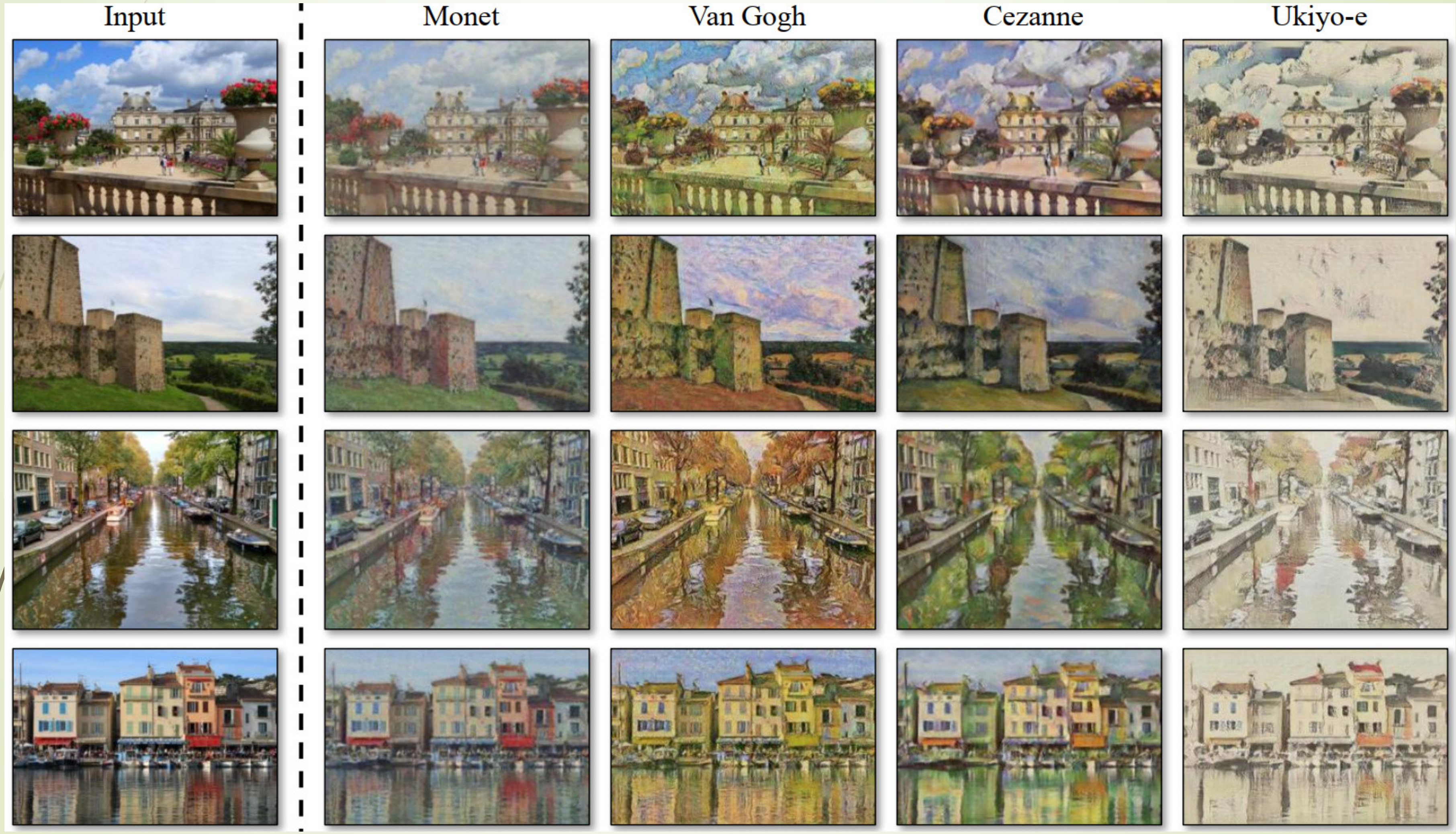
To preserve colors, we introduce:

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

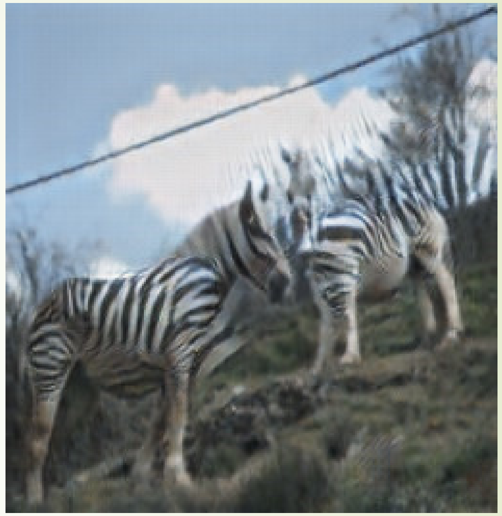


Monet to photographs

Photographs to different artists' styles



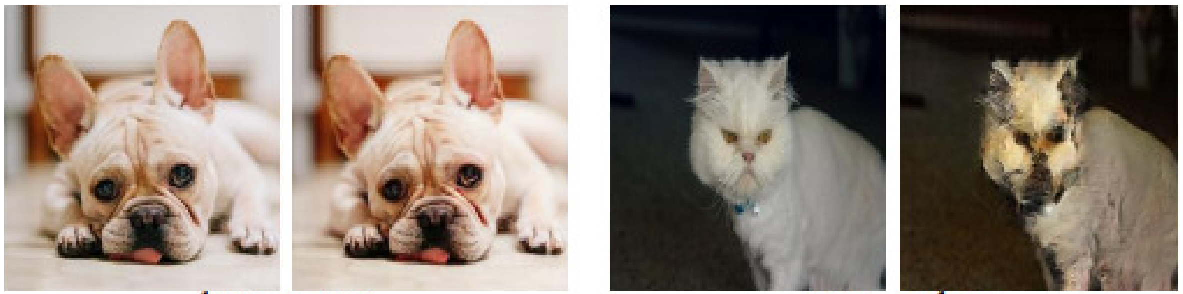
Object Transfiguration



More at

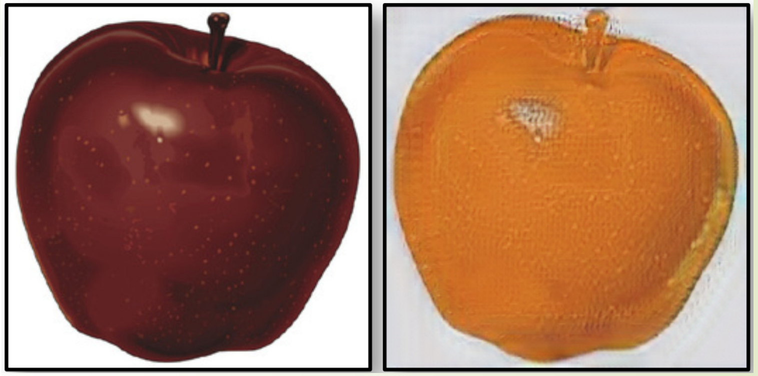
junyanz.github.io/CycleGAN

Limitations



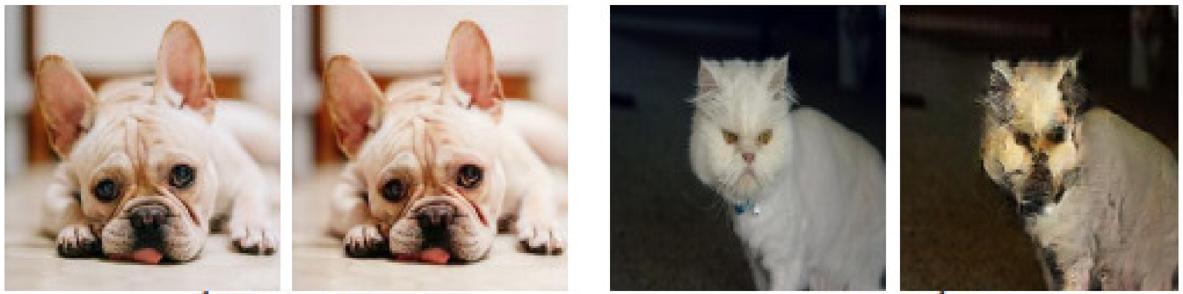
dog → cat

cat → dog

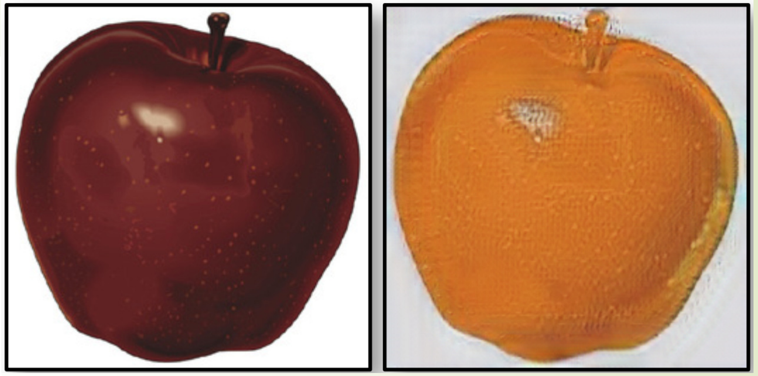


Limitations

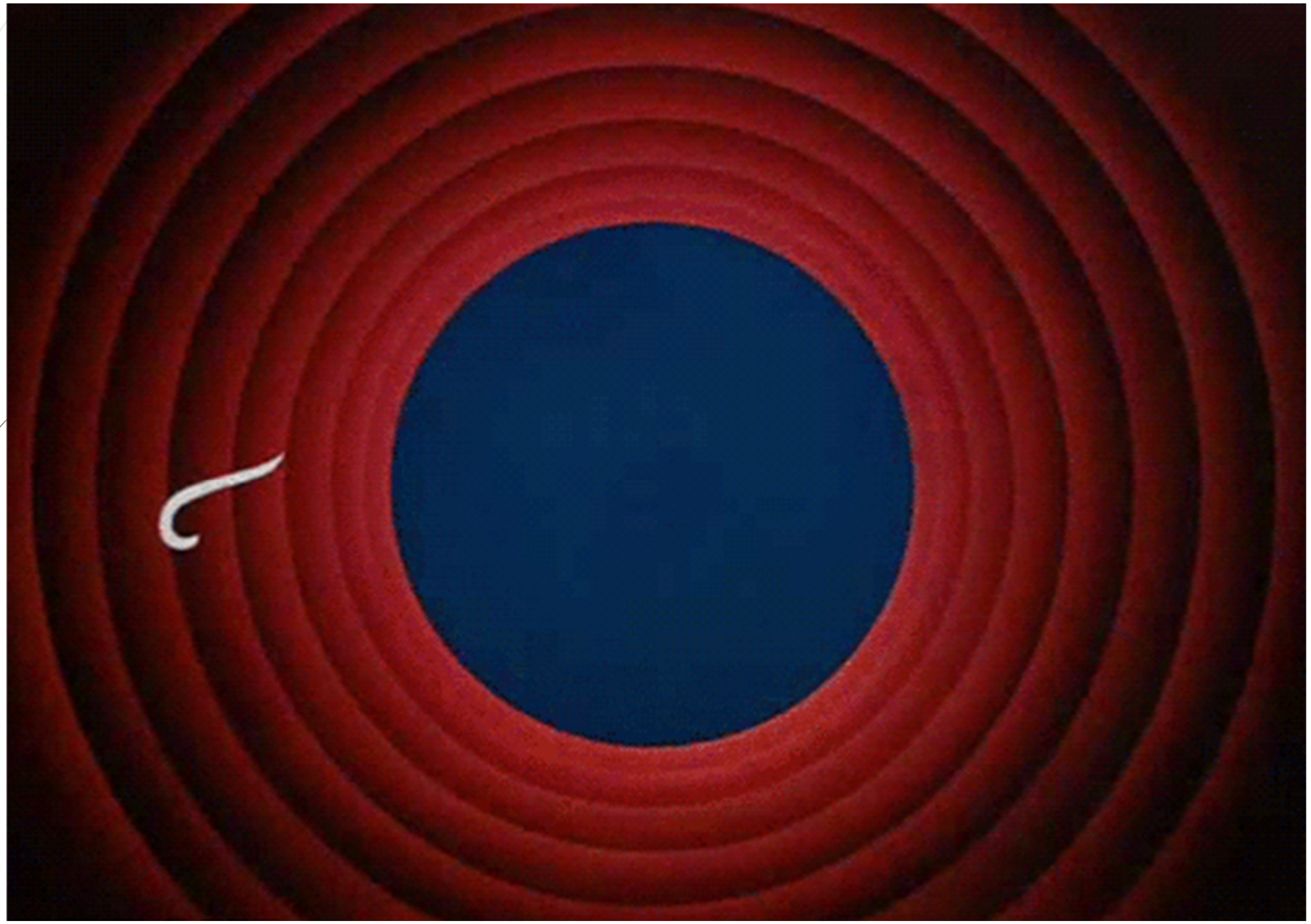
Geometric changes



dog → cat   cat → dog

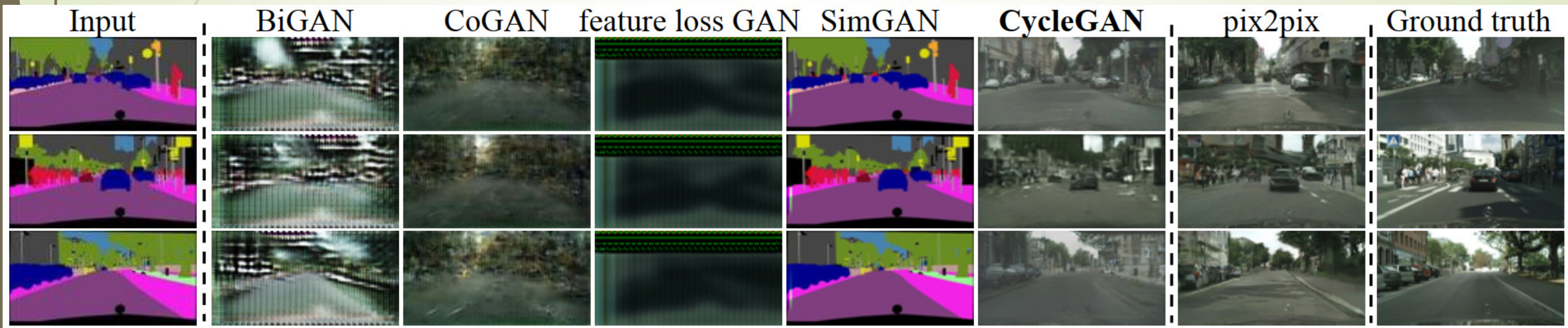


Inputs different from training data



Backup

results. In particular, for a GAN loss $\mathcal{L}_{\text{GAN}}(G, D, X, Y)$, we train the G to minimize $\mathbb{E}_{x \sim p_{\text{data}}(x)}[(D(G(x)) - 1)^2]$ and train the D to minimize $\mathbb{E}_{y \sim p_{\text{data}}(y)}[(D(y) - 1)^2] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[D(G(x))^2]$.


Input
BiGAN
CoGAN
feature loss GAN
SimGAN
CycleGAN
pix2pix
Ground truth

nators using a history of generated images rather than the ones produced by the latest generators. We keep an image buffer that stores the 50 previously created images.

For all the experiments, we set $\lambda = 10$ in Equation 3. We use the Adam solver [26] with a batch size of 1. All networks were trained from scratch with a learning rate of 0.0002. We keep the same learning rate for the first 100 epochs and linearly decay the rate to zero over the next 100 epochs. Please see the appendix (Section 7) for more details about the datasets, architectures, and training procedures.