Unpaired Image-to-Image Translation with CycleGAN

Keyur Patel, Khyatibahen Chaudhary, Manan Prajapati, Himen Sidhpura, Dhvani Agola, Jenny Mistry, Yash Sheth, Divyansh Thakar

Concordia University, Montreal

December 11, 2019

Paper

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017.

Style Transfer

Bank of Seine near Argenteuil, France

Style Transfer

Style Transfer

Style Transfer

Object Transfiguration

Object Transfiguration

Image-to-Image Translation

- Image-to-image translation is a class of vision and graphics problems
- Our goal is to to learn the mapping between an input image and an output image using a training set of aligned image pairs.

CNNs for Image-to-Image Translation

- CNNs are used in a wide variety of image prediction problems
- CNNs learn to minimize a loss function – used to score the quality of results
- But, What should we minimize?
- We take a naive approach and ask CNN to minimize Euclidean distance between predicted and ground truth pixels.
- But, it will produce blurry results.
- Coming up with loss functions that does what we really want - e.g., sharp and realistic images - is hard.

CNNs for Image-to-Image Translation

- CNNs are used in a wide variety of image prediction problems
- CNNs learn to minimize a loss function – used to score the quality of results
- But, What should we minimize?
- We take a naive approach and ask CNN to minimize Euclidean distance between predicted and ground truth pixels.
- But, it will produce blurry results.
- Coming up with loss functions that does what we really want - e.g., sharp and realistic images - is hard.

CNNs for Image-to-Image Translation

- CNNs are used in a wide variety of image prediction problems
- CNNs learn to minimize a loss function – used to score the quality of results
- But, What should we minimize?
- We take a naive approach and ask CNN to minimize Euclidean distance between predicted and ground truth pixels.
- But, it will produce blurry results.
- Coming up with loss functions that does what we really want - e.g., sharp and realistic images - is hard.

CNNs for Image-to-Image Translation

- CNNs are used in a wide variety of image prediction problems
- CNNs learn to minimize a loss function – used to score the quality of results
- But, What should we minimize?
- We take a naive approach and ask CNN to minimize Euclidean distance between predicted and ground truth pixels.
- But, it will produce blurry results.
- Coming up with loss functions that does what we really want - e.g., sharp and realistic images - is hard.

CNNs for Image-to-Image Translation

- CNNs are used in a wide variety of image prediction problems
- CNNs learn to minimize a loss function – used to score the quality of results
- But, What should we minimize?
- We take a naive approach and ask CNN to minimize Euclidean distance between predicted and ground truth pixels.
- But, it will produce blurry results.
- Coming up with loss functions that does what we really want - e.g., sharp and realistic images - is hard.

CNNs for Image-to-Image Translation

- CNNs are used in a wide variety of image prediction problems
- CNNs learn to minimize a loss function – used to score the quality of results
- But, What should we minimize?
- We take a naive approach and ask CNN to minimize Euclidean distance between predicted and ground truth pixels.
- But, it will produce blurry results.
- Coming up with loss functions that does what we really want - e.g., sharp and realistic images - is hard.

Generative Adversarial Networks

- GANs allows to specify high-level goal. For example, "make the generated images indistinguishable from real images".
- GANs simultaneously learn two models: a discriminative model D that tries to classify if the output image is real or fake, and a generative model G that captures the data distribution.
- Two neural networks contest with each other in a game.
- Yann LeCun described GANs as "the coolest idea in machine learning in the last twenty years".

Generative Adversarial Networks

- GANs allows to specify high-level goal. For example, "make the generated images indistinguishable from real images".
- GANs simultaneously learn two models: a discriminative model D that tries to classify if the output image is real or fake, and a generative model G that captures the data distribution.
- Two neural networks contest with each other in a game.
- Yann LeCun described GANs as "the coolest idea in machine learning in the last twenty years".

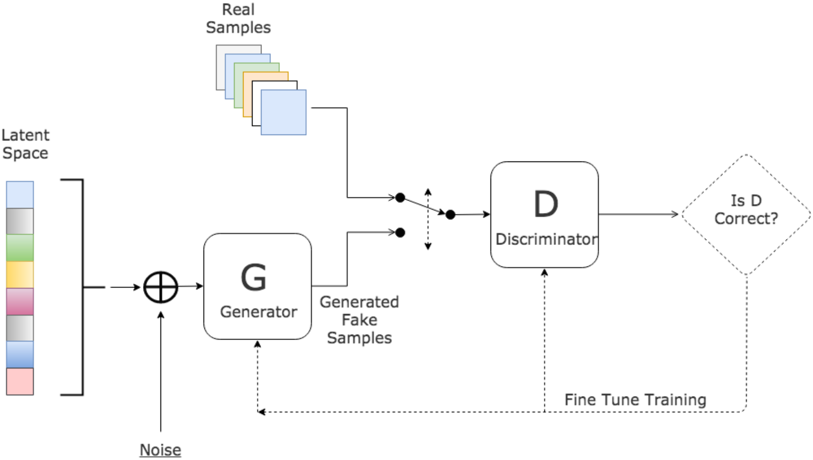
Generative Adversarial Networks

- GANs allows to specify high-level goal. For example, "make the generated images indistinguishable from real images".
- GANs simultaneously learn two models: a discriminative model D that tries to classify if the output image is real or fake, and a generative model G that captures the data distribution.
- Two neural networks contest with each other in a game.
- Yann LeCun described GANs as "the coolest idea in machine learning in the last twenty years".

Generative Adversarial Networks

- GANs allows to specify high-level goal. For example, "make the generated images indistinguishable from real images".
- GANs simultaneously learn two models: a discriminative model D that tries to classify if the output image is real or fake, and a generative model G that captures the data distribution.
- Two neural networks contest with each other in a game.
- Yann LeCun described GANs as "the coolest idea in machine learning in the last twenty years".

Generative Adversarial Networks



Math behind GANs

- Learn a generator's distribution p_g over data x:

$$x = G(\boldsymbol{z}; \theta_g) \tag{1}$$

  p_z is distribution over noise input z. G is represented by multilayer perceptron with parameters θ_g.

- Similarly, We define another multilayer perceptron $D(\boldsymbol{x}; \theta_d)$ that represents the probability that x is from the real data, not p_g.

Math behind GANs

- Learn a generator's distribution p_g over data x:

$$x = G(\boldsymbol{z}; \theta_g) \tag{1}$$

  p_z is distribution over noise input z. G is represented by multilayer perceptron with parameters θ_g.

- Similarly, We define another multilayer perceptron $D(\boldsymbol{x}; \theta_d)$ that represents the probability that x is from the real data, not p_g.

Math behind GANs

- Given a real data, we want D to maximize $\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]$.
- Meanwhile, given a fake sample $G(z)$, we want D to maximize $\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$ where $D(G(\boldsymbol{z}))$ is the discriminator's estimate of the probability that a fake instance is real..
- However, G is trained to minimize $\log(1 - D(G(\boldsymbol{z})))$ simultaneously.

Math behind GANs

- Given a real data, we want D to maximize $\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]$.
- Meanwhile, given a fake sample $G(z)$, we want D to maximize $\mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$ where $D(G(\boldsymbol{z}))$ is the discriminator's estimate of the probability that a fake instance is real..
- However, G is trained to minimize $\log(1 - D(G(\boldsymbol{z})))$ simultaneously.

Math behind GANs

- Given a real data, we want D to maximize $\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]$.
- Meanwhile, given a fake sample $G(z)$, we want D to maximize $\mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$ where $D(G(\boldsymbol{z}))$ is the discriminator's estimate of the probability that a fake instance is real..
- However, G is trained to minimize $\log(1 - D(G(\boldsymbol{z})))$ simultaneously.

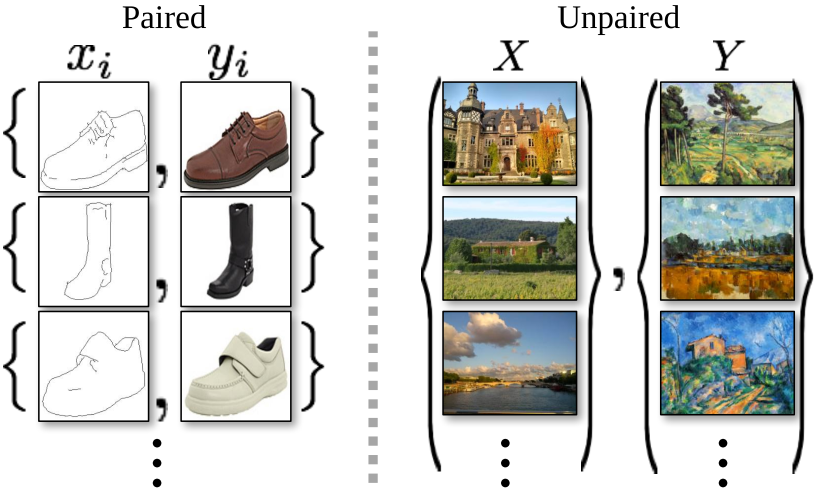
Loss Function

In short, G and D are playing zero-sum game with following loss function:

$$\min_{G}\max_{D} V(D, G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{2}$$

Recent Studies

Unpaired Image-to-Image Translation



Problems with Paired Image-to-Image Translation

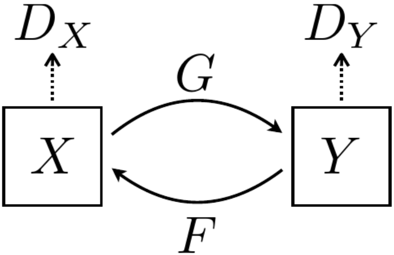
- Paired training data is difficult to obtain
- For object transfiguration (e.g., zebra$\leftrightarrow$horse), the desired output is not even well-defined

Formulation

Goal: Learn mapping functions between two domains X and Y given training samples $\{x_i\}_{i=1}^{N}$ where $x_i \in X$ and $\{y_j\}_{j=1}^{M}$ where $y_j \in Y$. We denote the data distribution as $x \sim p_{data}(x)$ and $y \sim p_{data}(y)$.

Adversarial Loss

Model includes two mappings $G : X \to Y$ and $F : Y \to X$. In addition to that, two adversarial discriminators D_X and D_Y, where D_X aims to distinguish between images $\{x\}$ and translated images $\{F(y)\}$; in the same way, D_Y aims to discriminate between $\{y\}$ and $\{G(x)\}$.



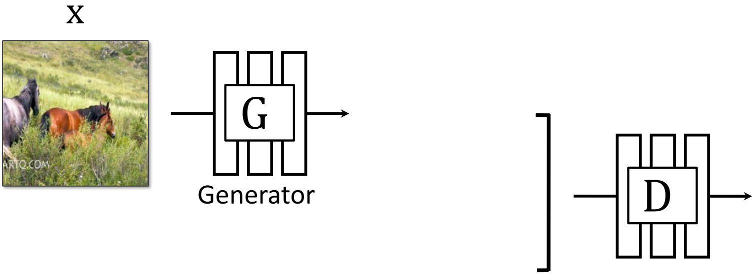
Adversarial Loss

For the mapping function $G : X \rightarrow Y$ and its discriminator D_Y, we can write adversarial loss as:

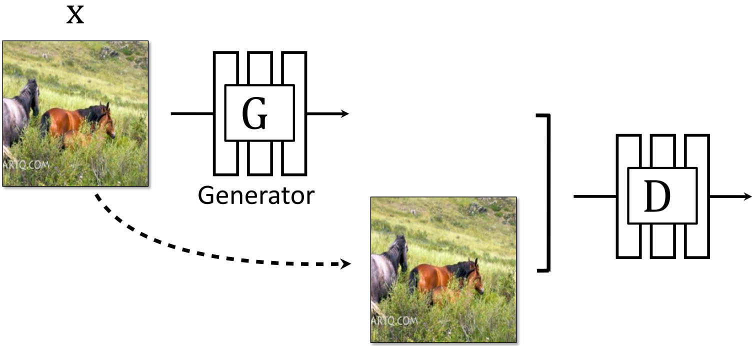
$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))], \quad (3)$$

where G will try to generate images indistinguishable from Y, on the other hand D_y learns to distinguish between fake and real images.

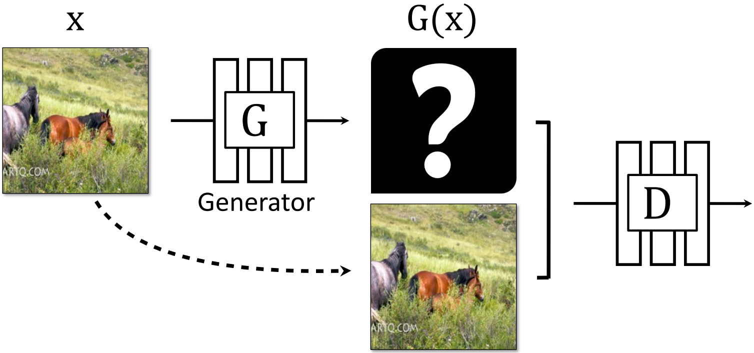
Mode Collapse



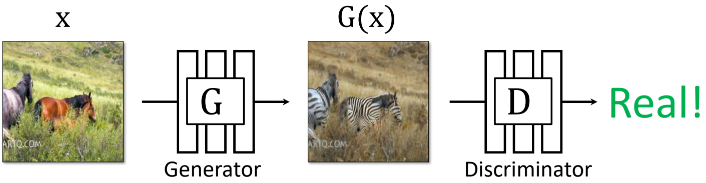
Mode Collapse



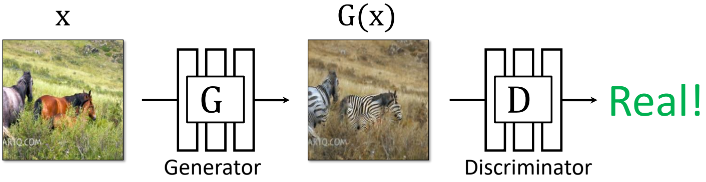
Mode Collapse

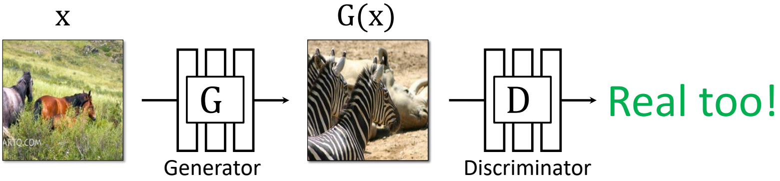


Mode Collapse



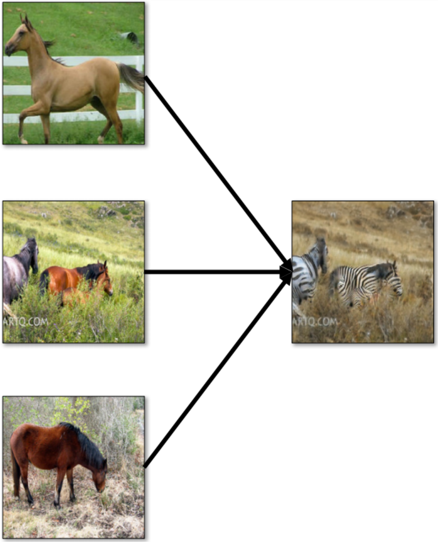
Mode Collapse





Mode Collapse

Mode Collapse!
GANs do **not** force output to correspond to input.



Cycle Consistency Loss

It's a beautiful day

C'est une belle journée

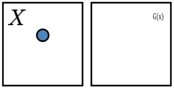
It's a beautiful day

Cycle Consistency Loss

x

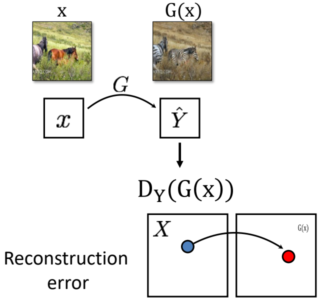
$\boxed{x}$

Reconstruction error



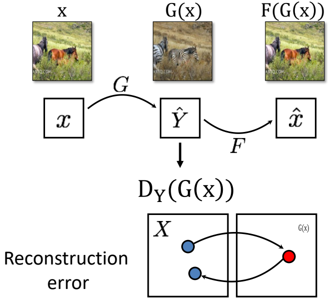
[Zhu*, Park*, Isola, and Efros, ICCV 2017]

Cycle Consistency Loss



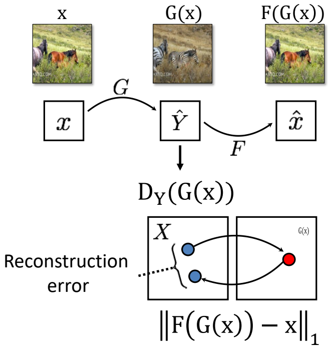
[Zhu*, Park*, Isola, and Efros, ICCV 2017]

Cycle Consistency Loss



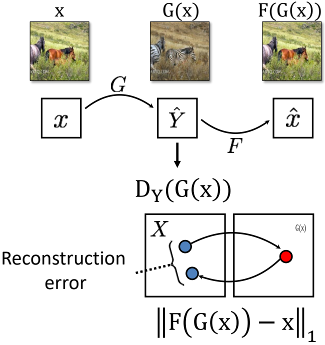
[Zhu*, Park*, Isola, and Efros, ICCV 2017]

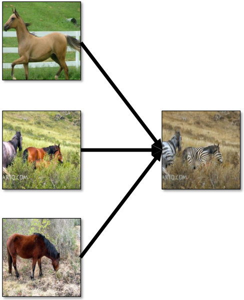
Cycle Consistency Loss



[Zhu*, Park*, Isola, and Efros, ICCV 2017]

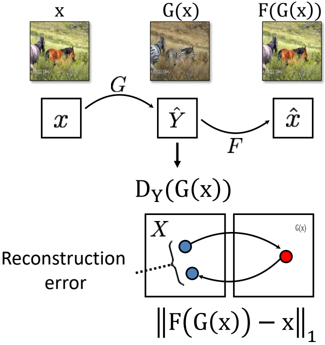
Cycle Consistency Loss

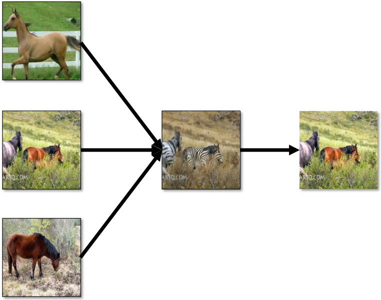




[Zhu*, Park*, Isola, and Efros, ICCV 2017]

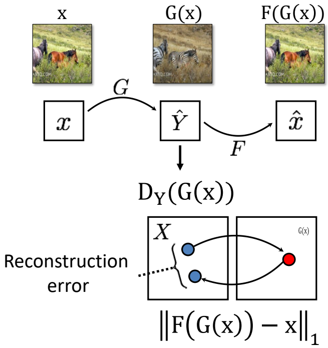
Cycle Consistency Loss



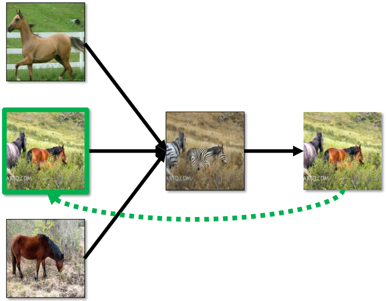


[Zhu*, Park*, Isola, and Efros, ICCV 2017]

Cycle Consistency Loss

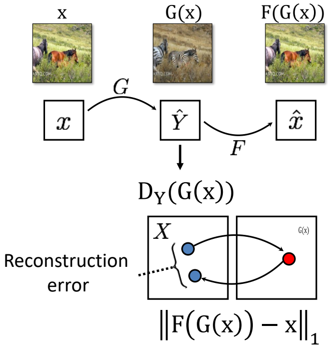


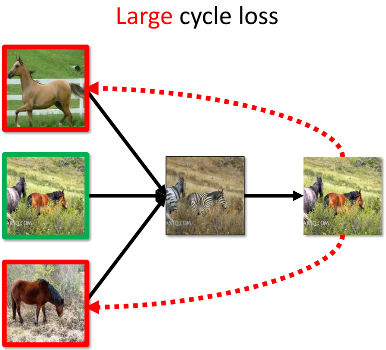
Small cycle loss



[Zhu*, Park*, Isola, and Efros, ICCV 2017]

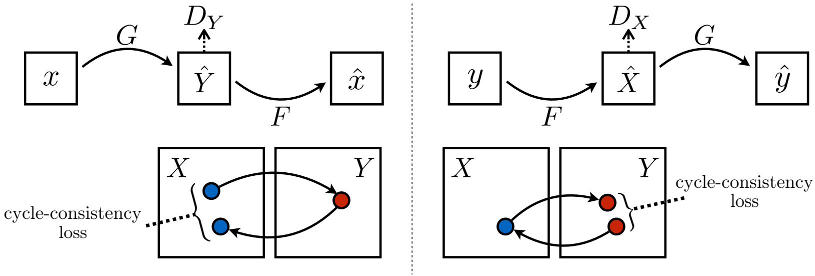
Cycle Consistency Loss





[Zhu*, Park*, Isola, and Efros, ICCV 2017]

Cycle Consistency Loss



For each image x from domain X, the image translation cycle should be able to bring x back to the original image, i.e., $x \to G(x) \to F(G(x)) \approx x$. This is called *forward cycle consistency*.

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = &\ \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] \\ &+ \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1], \end{aligned} \tag{4}$$

Full Objective

Our full objective is:

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) =& \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ &+ \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ &+ \lambda \mathcal{L}_{\text{cyc}}(G, F),\end{aligned} \tag{5}$$

Results

Results

Results

Results

Results

Results

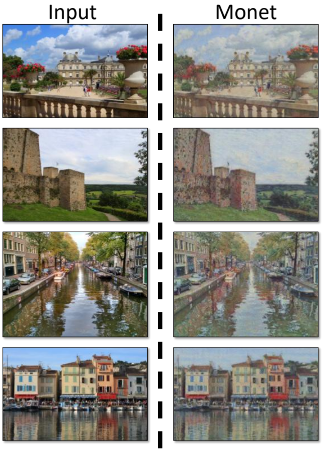
Results

Results

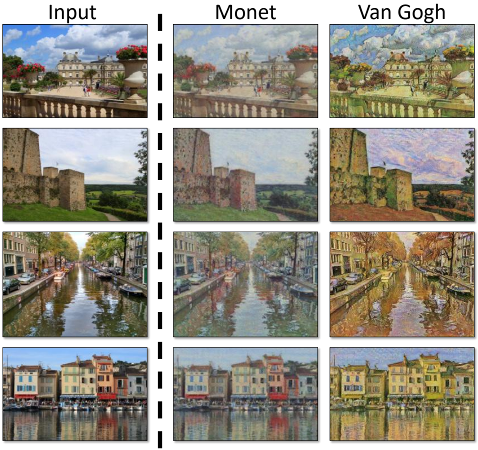
Results

Input

Results



Results

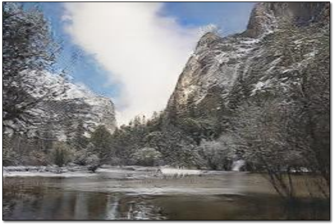


Results

Results

Results

Results



Results

Results

Why it works?

Adversarial Loss : Change the Style

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

Cycle Consistency Loss: Preserve the Original Content

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

Failure Cases

Failure Cases

Failure Cases

- Caused by the distribution characteristics of the training datasets
- Fails at varied and extreme transformations, especially geometric changes

Demo

Season Transfer

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017.