Literature review and Implementation of Style Transfer

UIC 统计系

李杰

1730005024

What’s style transfer?

- Computer version-Image transfer-**Style-transfer**

Literature review of Style-Transfer

A method of object recognition:

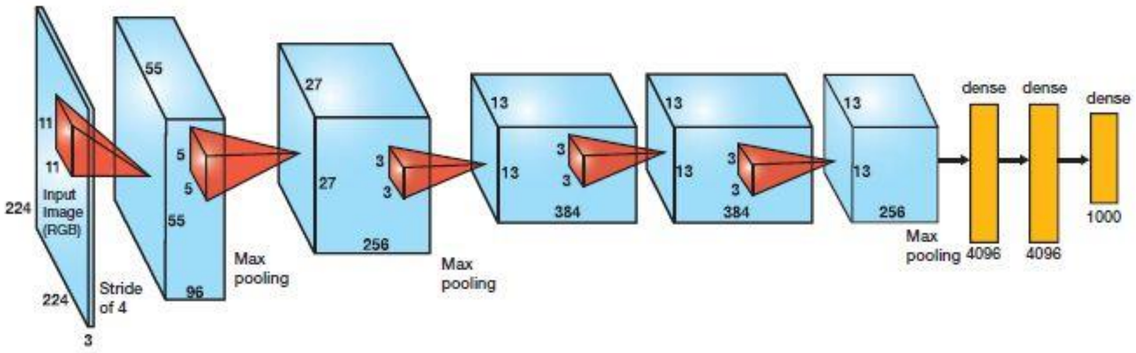
"Bag of feature"

The object we want to recognition

Local feature (select by human)

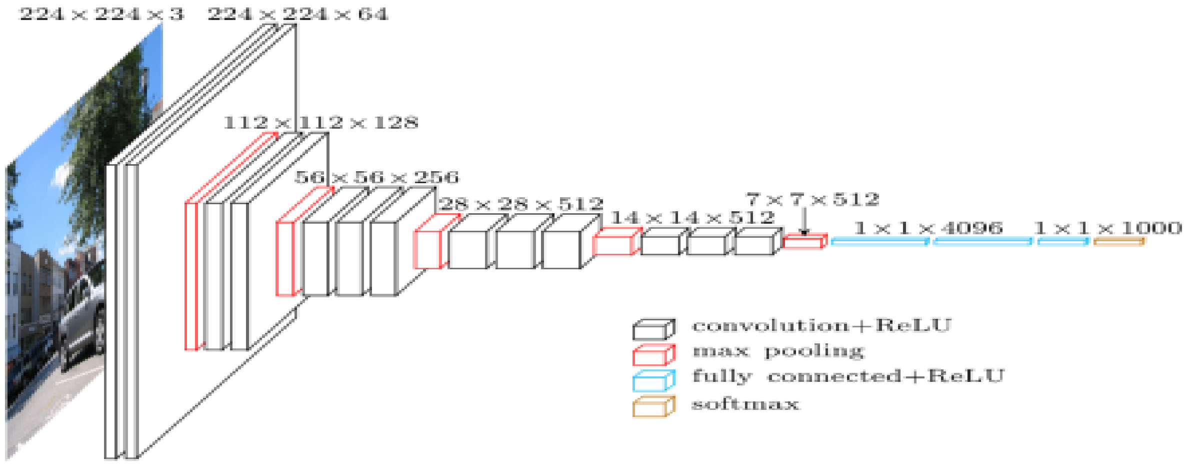
Literature review of Style-Transfer

How to do the object recognition better?



AlexNet(2012)

VGG16(2014)



Literature review of Style-Transfer

- **How to use small kernel(VGG16) to replace large kernel(Alex-net) ?**

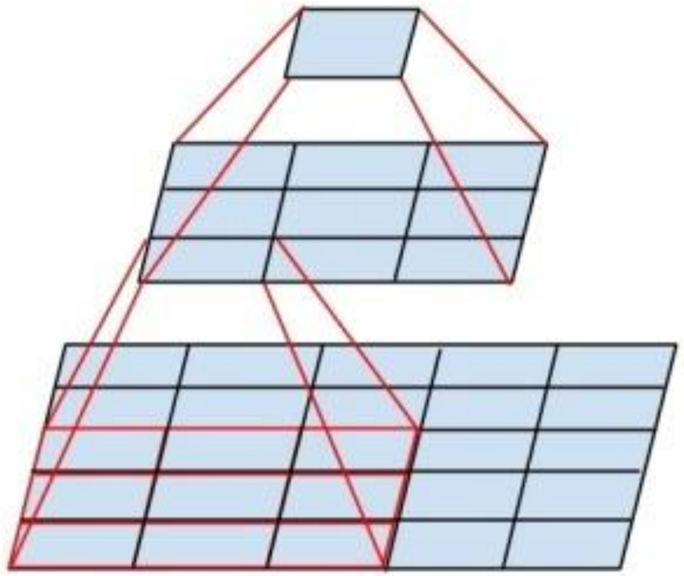


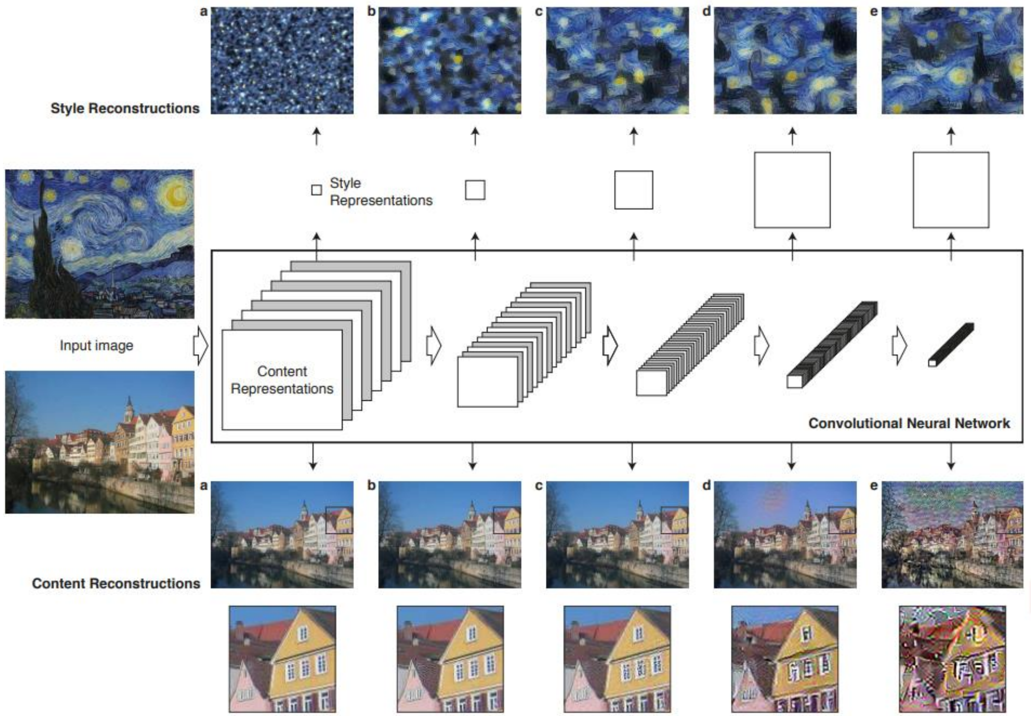
Figure 1. Mini-network replacing the 5 × 5 convolutions.

Literature review of Style-Transfer

- **2015: The first model**

A Neural Algorithm of Artistic Style

Leon A. Gatys,[1,2,3*] Alexander S. Ecker,[1,2,4,5] Matthias Bethge[1,2,4]



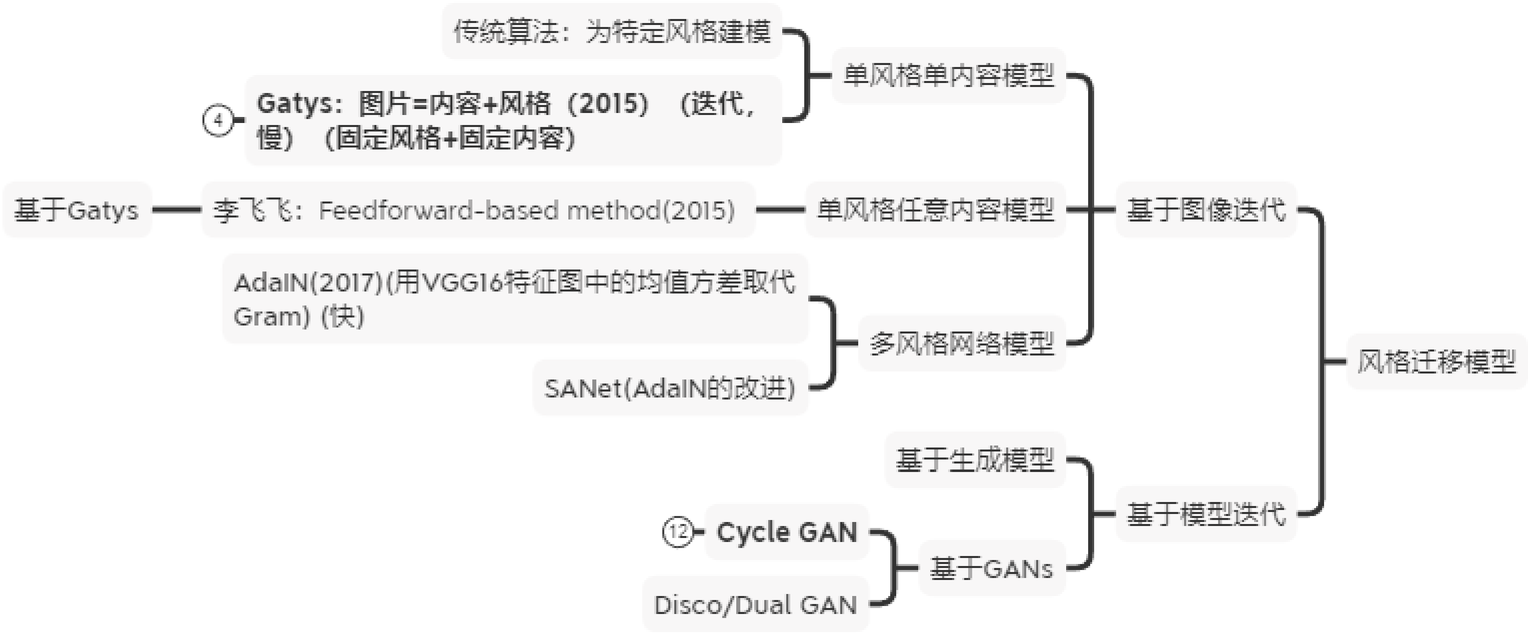
Use VGG16 to extract the feature of **Content** and **Style** from two pictures

基于深度学习的图像风格迁移研究综述*

陈淑環[†]，韦玉科，徐 乐，董晓华，温坤哲
（广东工业大学 计算机学院，广州 510006）

Literature review of Style-Transfer

- **Other important model**



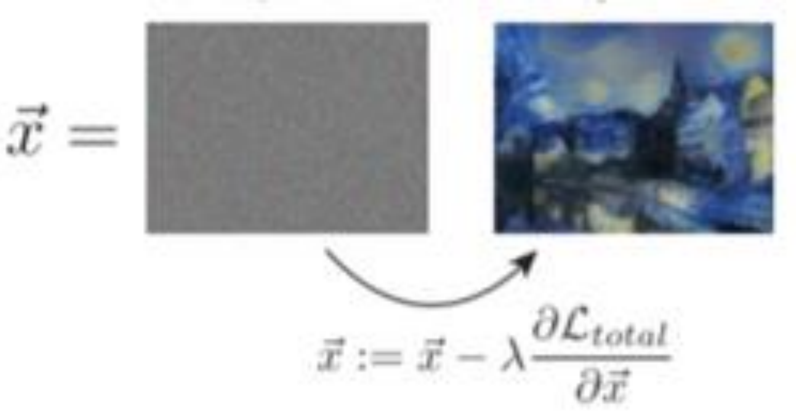
A Neural Algorithm of Artistic Style

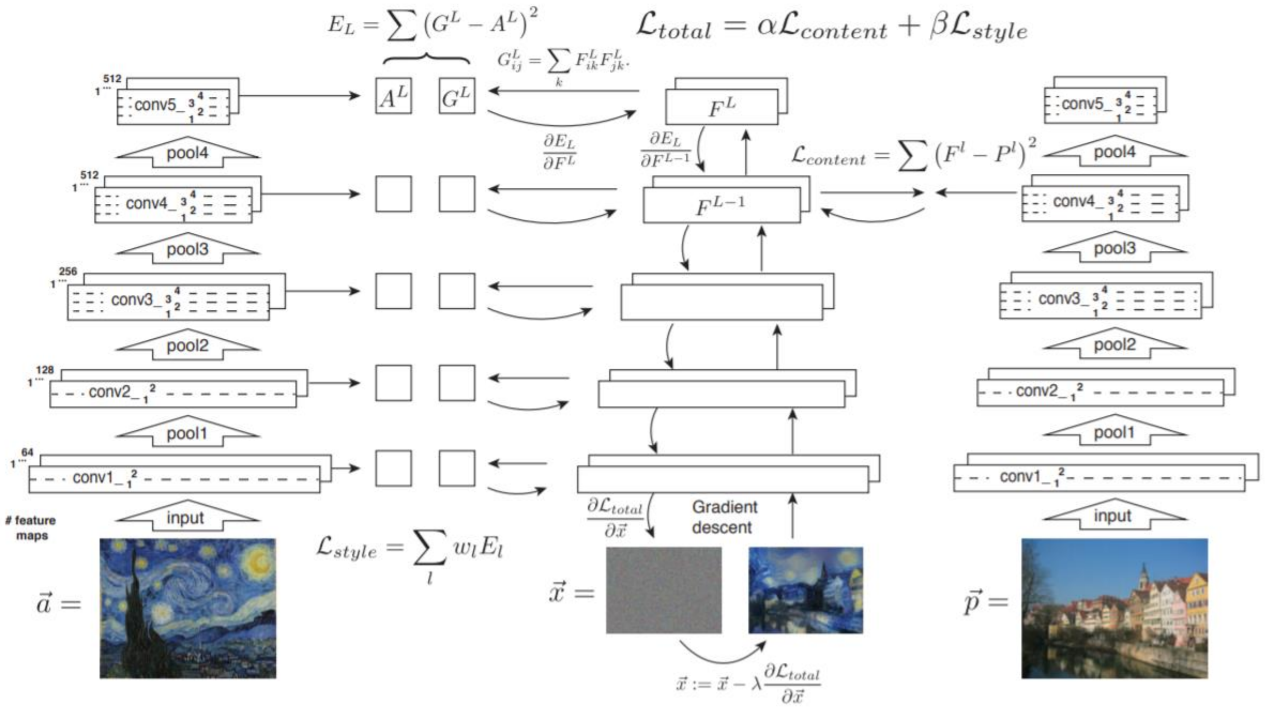
Leon A. Gatys,[1,2,3*] Alexander S. Ecker,[1,2,4,5] Matthias Bethge[1,2,4]

Gaty's model

Two assumption:
Image = Content+Style
Style = Texture

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style}$$





A Neural Algorithm of Artistic Style

Leon A. Gatys,[1,2,3*] Alexander S. Ecker,[1,2,4,5] Matthias Bethge[1,2,4]

Gaty's model

Measure the content:

$$\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}, l) = \frac{1}{2}\sum_{i,j}\left(F_{ij}^{l} - P_{ij}^{l}\right)^2 . \qquad (1)$$

Represent the respond of content in Layer l (l=1~Nl), Filter i(l = 1~Ml) , Position j

The derivative of this loss with respect to the activations in layer l equals

$$\frac{\partial \mathcal{L}_{\text{content}}}{\partial F_{ij}^{l}} = \begin{cases} \left(F^{l} - P^{l}\right)_{ij} & \text{if } F_{ij}^{l} > 0 \\ 0 & \text{if } F_{ij}^{l} < 0 , \end{cases} \qquad (2)$$

A Neural Algorithm of Artistic Style

Leon A. Gatys,[1,2,3*] Alexander S. Ecker,[1,2,4,5] Matthias Bethge[1,2,4]

Gaty's model

Measure the style:
suppose that the texture of image can represent the style

Gram matrix: used to capture the texture information of Image $G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l.$

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^l - A_{ij}^l\right)^2$$

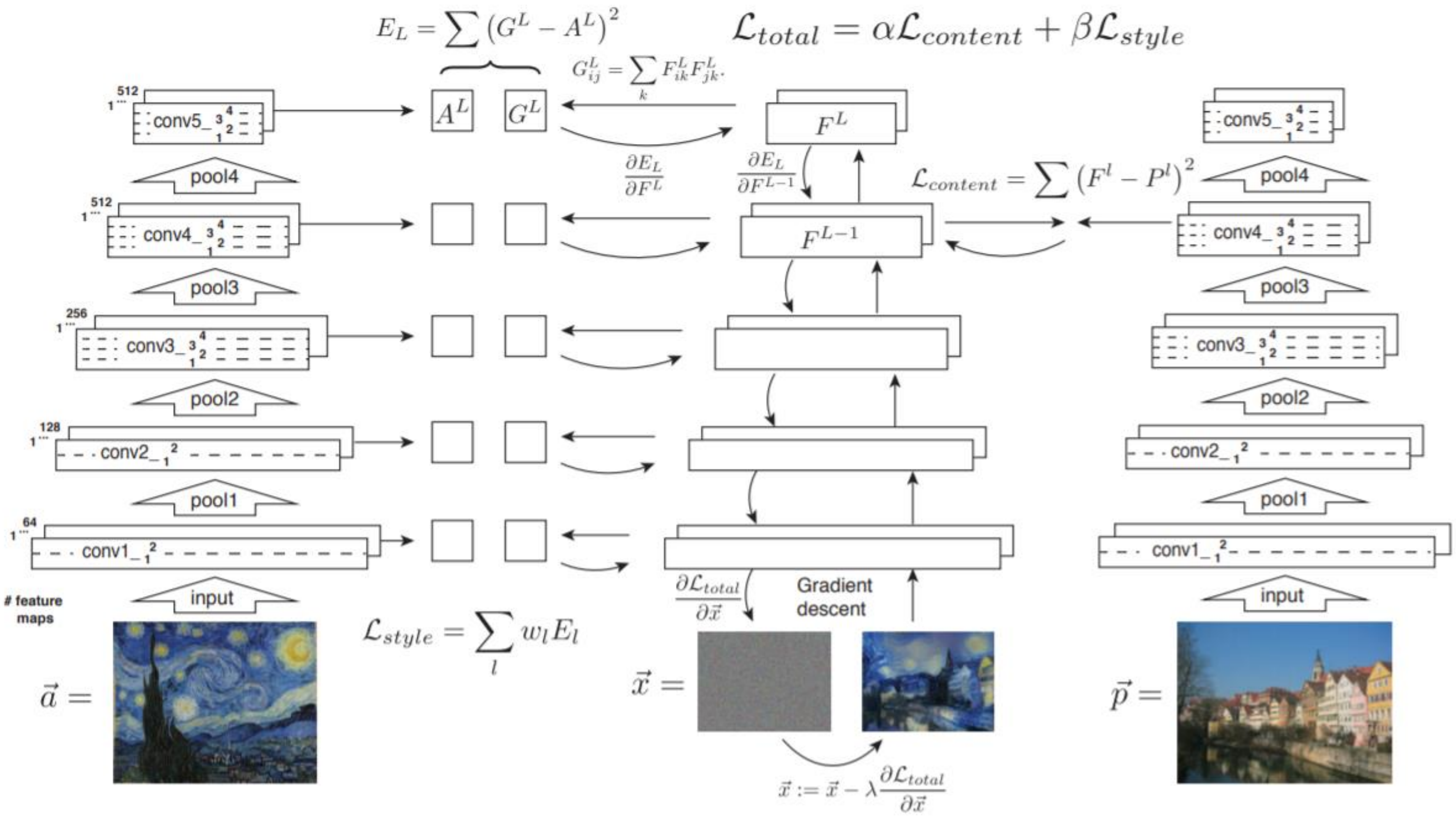
$$\mathcal{L}_{\text{style}}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l,$$

$$\frac{\partial E_l}{\partial F_{ij}^l} = \begin{cases} \frac{1}{N_l^2 M_l^2} \left(\left(F^l\right)^{\mathrm{T}} \left(G^l - A^l\right)\right)_{ji} & \text{if } F_{ij}^l > 0 \\ 0 & \text{if } F_{ij}^l < 0 \end{cases}$$

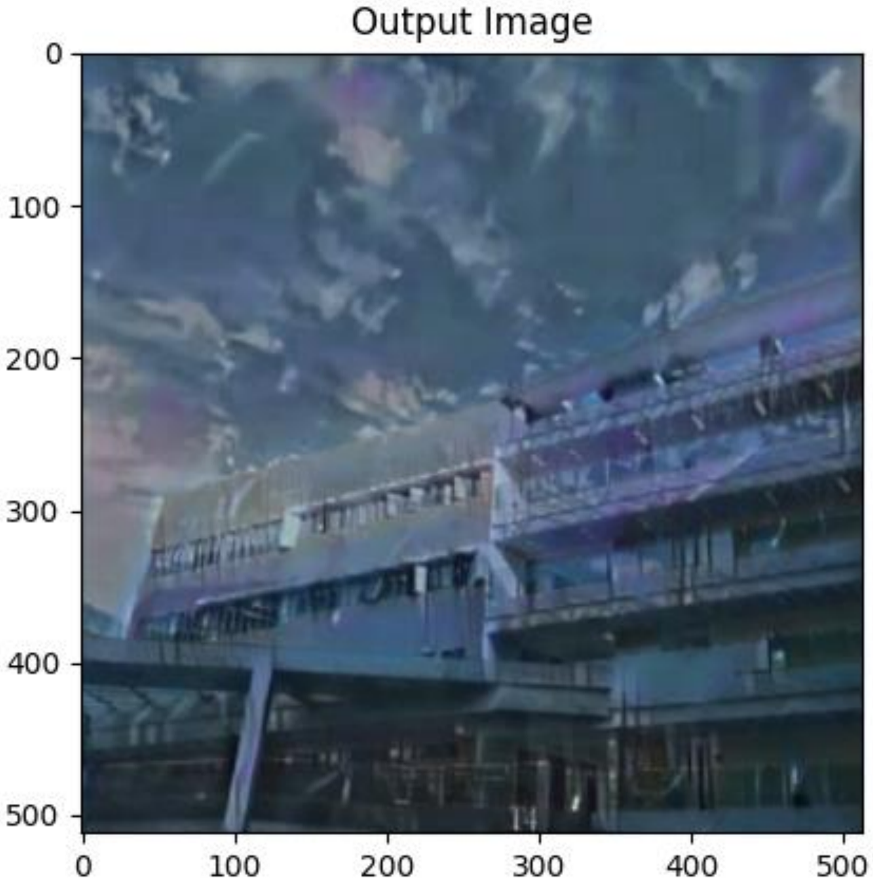
A Neural Algorithm of Artistic Style

Leon A. Gatys,[1,2,3*] Alexander S. Ecker,[1,2,4,5] Matthias Bethge[1,2,4]

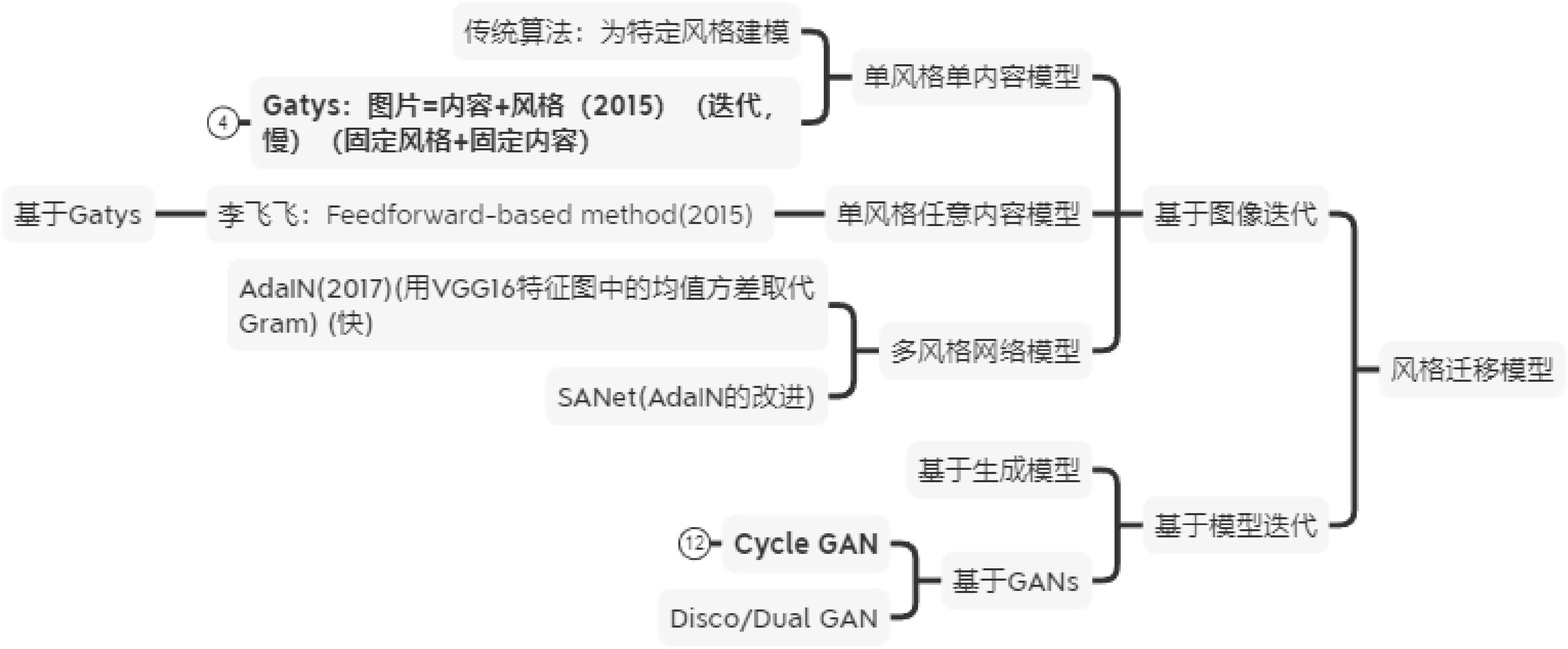
Gaty's model



Implementation of Gaty’s Model (Base on Pytorch)



Style Transfer base on Cycle GANs



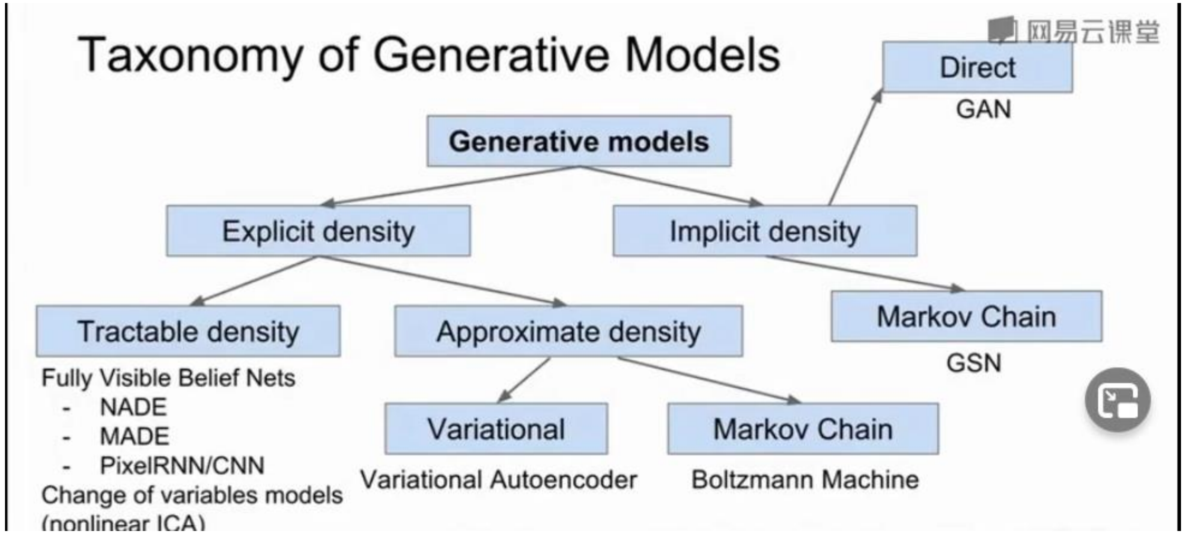
Generate Model

Given training data, generate new samples from same distribution

Training data ~ $p_{data}(x)$

Generated samples ~ $p_{model}(x)$

Want to learn $p_{model}(x)$ similar to $p_{data}(x)$

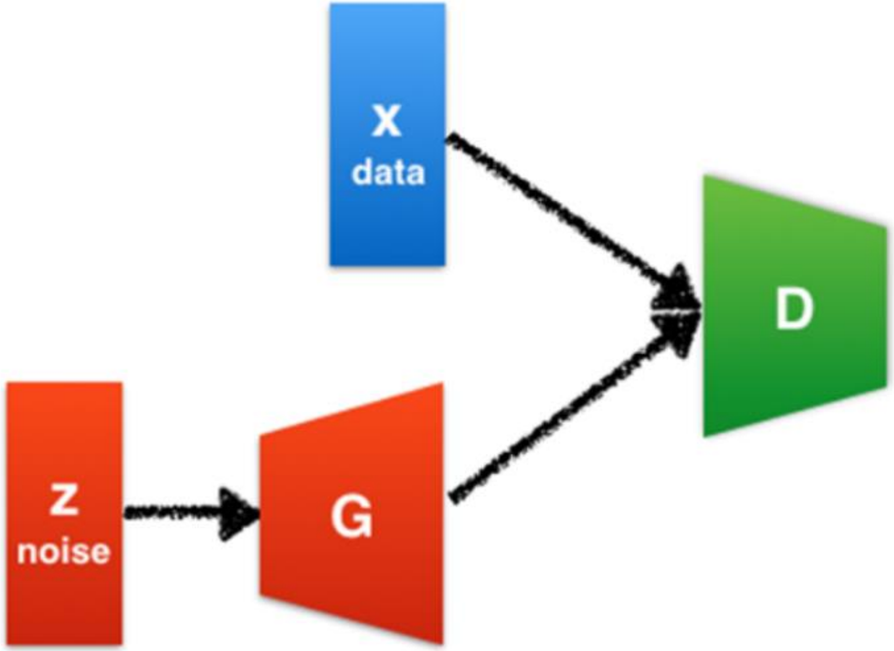


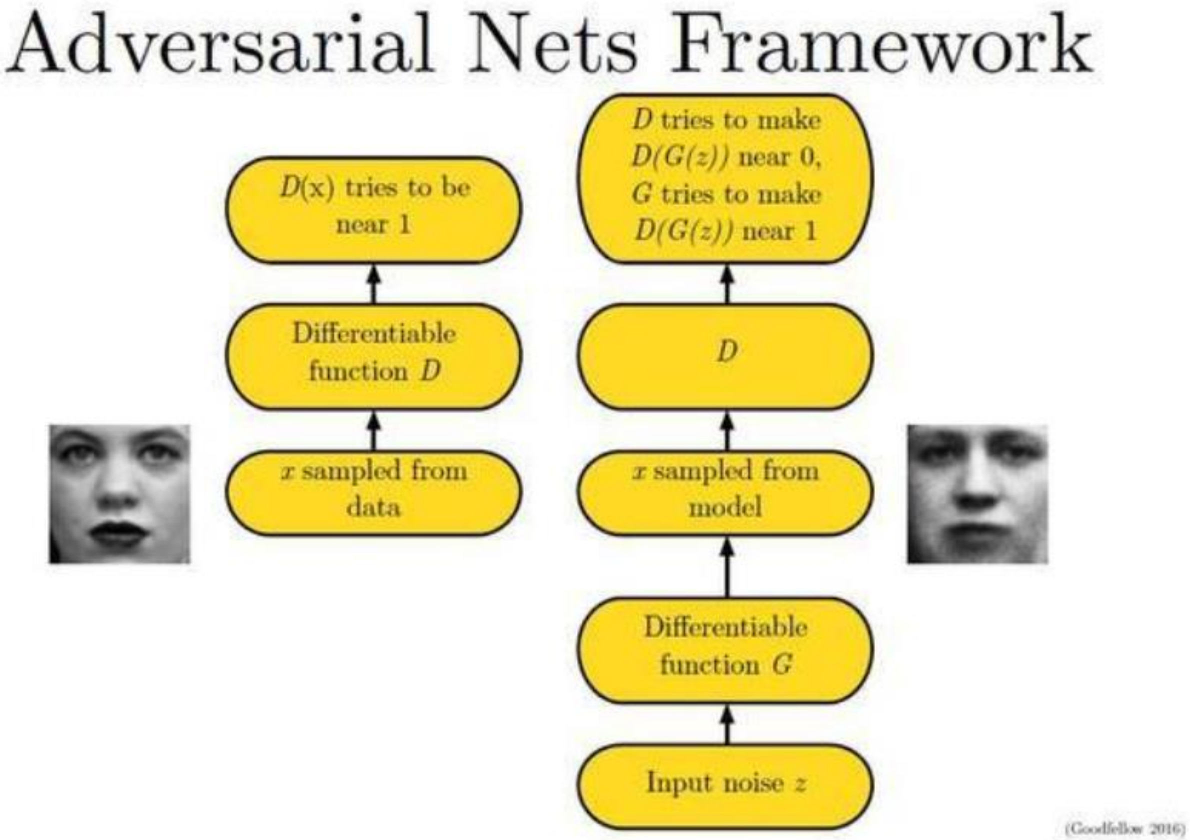
Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie,[*] Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,[†] Aaron Courville, Yoshua Bengio[‡]
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

GANs（Ian J.Goodfellow 2014）

(对抗生成网络)





Generative Adversarial Nets

Ian J. Goodfellow, Jean Pouget-Abadie,* Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair,† Aaron Courville, Yoshua Bengio‡
Département d'informatique et de recherche opérationnelle
Université de Montréal
Montréal, QC H3C 3J7

GANs（Ian J.Goodfellow 2014）

(对抗生成网络)

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

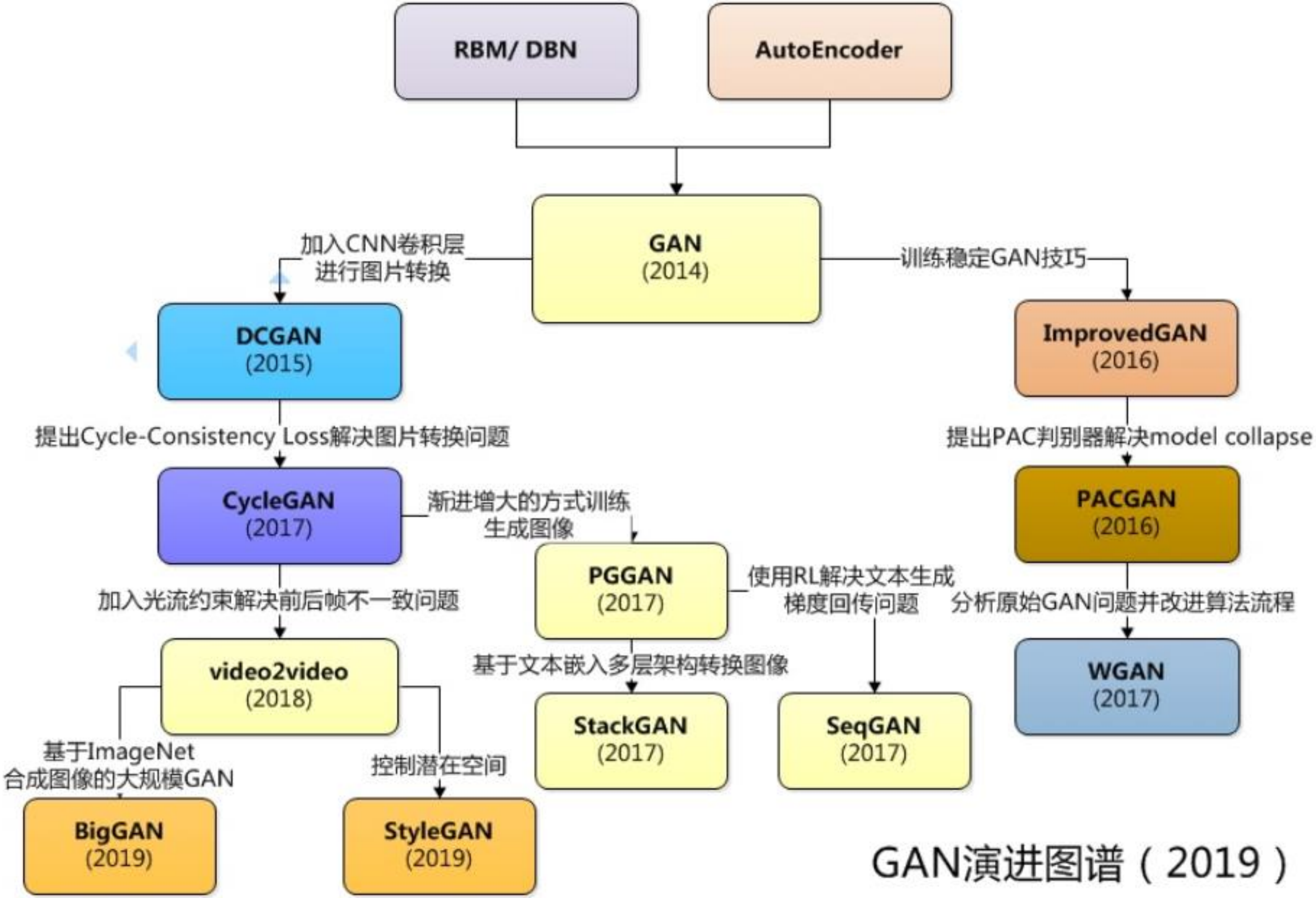
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

$$\nabla C \approx \frac{1}{m} \sum_{j=1}^{m} \nabla C_{X_j},$$



GAN演进图谱（2019）

DCGAN(Deep Convolutional GAN):
Use two CNN as D&G

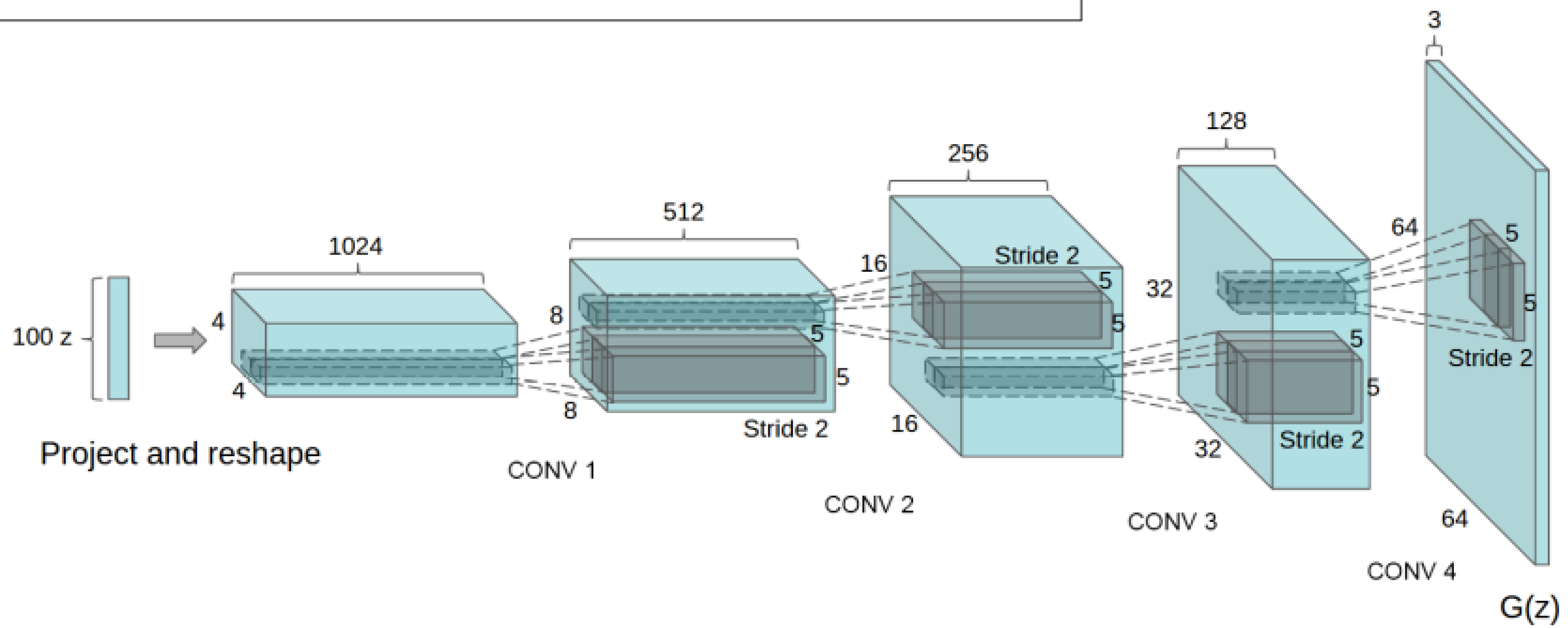
UNSUPERVISED REPRESENTATION LEARNING
WITH DEEP CONVOLUTIONAL
GENERATIVE ADVERSARIAL NETWORKS

Alec Radford & Luke Metz
indico Research
Boston, MA
{alec,luke}@indico.io

Soumith Chintala
Facebook AI Research
New York, NY
soumith@fb.com

Architecture guidelines for stable Deep Convolutional GANs

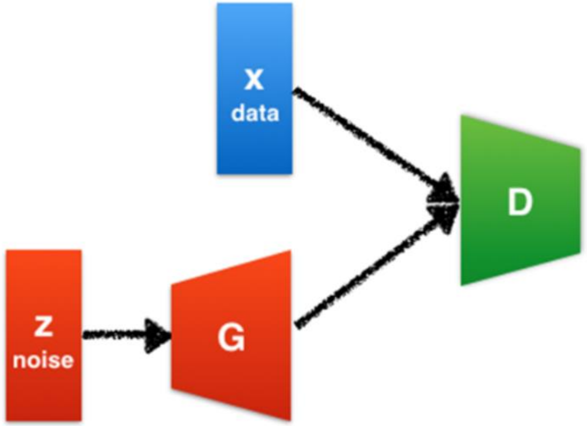
- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

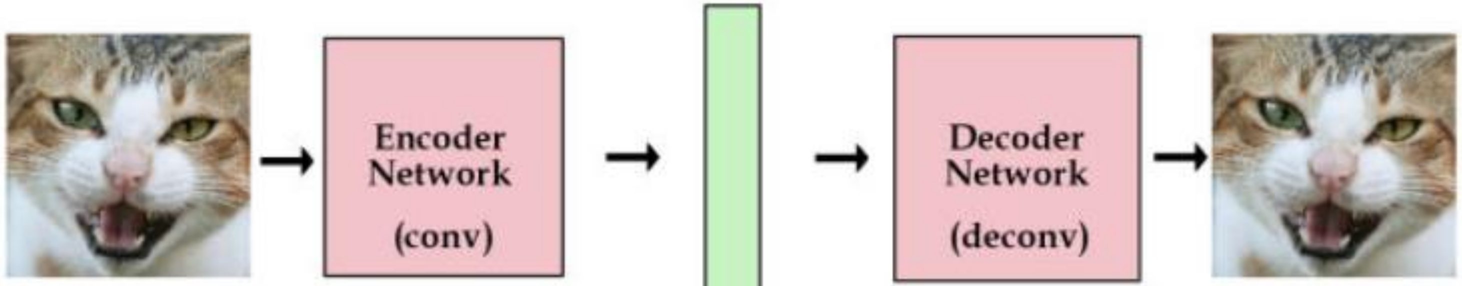


Why cycle-GANs?

Reason 1: Input for GANs is white noise

Recall the **Purpose of Style transfer** and **GANs**

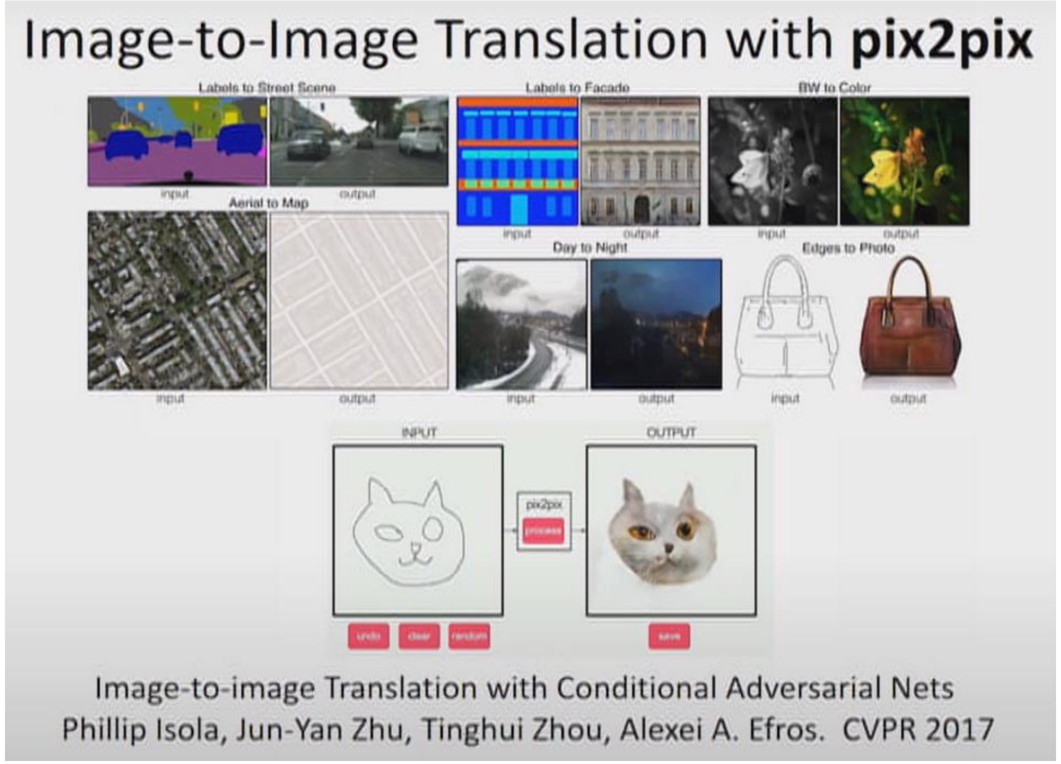


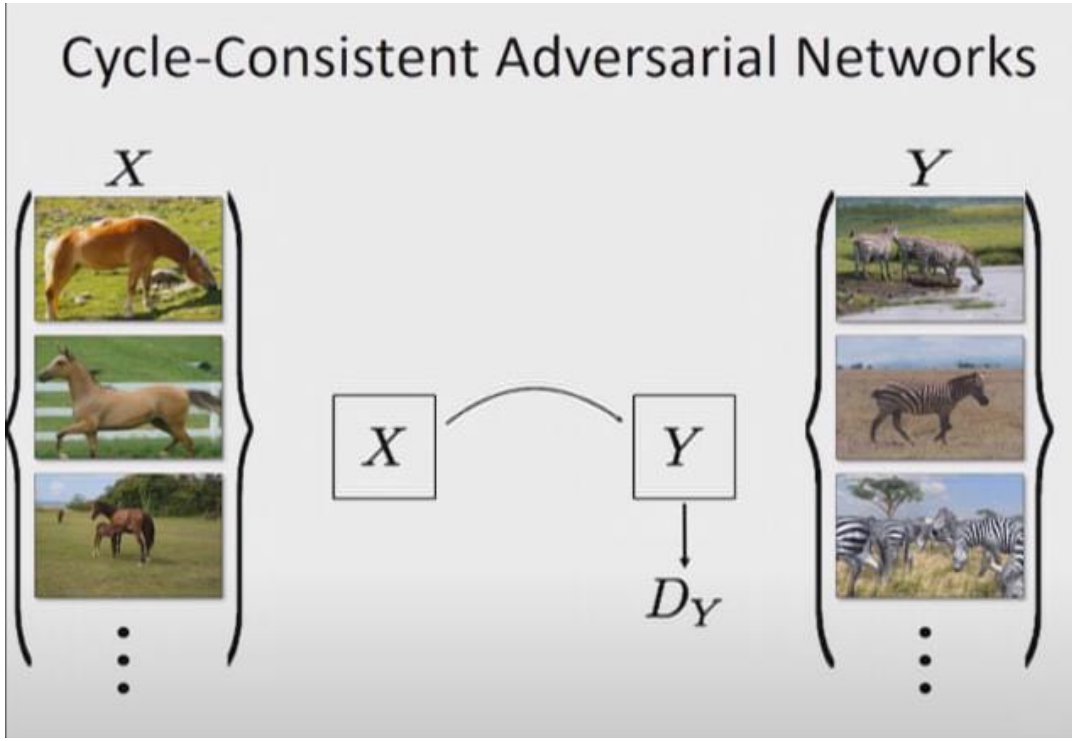


(Input Image) latent vector / variables (Gnerated Image)

Why cycle-GANs?

Reason 2: Lack of pair data, the output will be not contain the same content

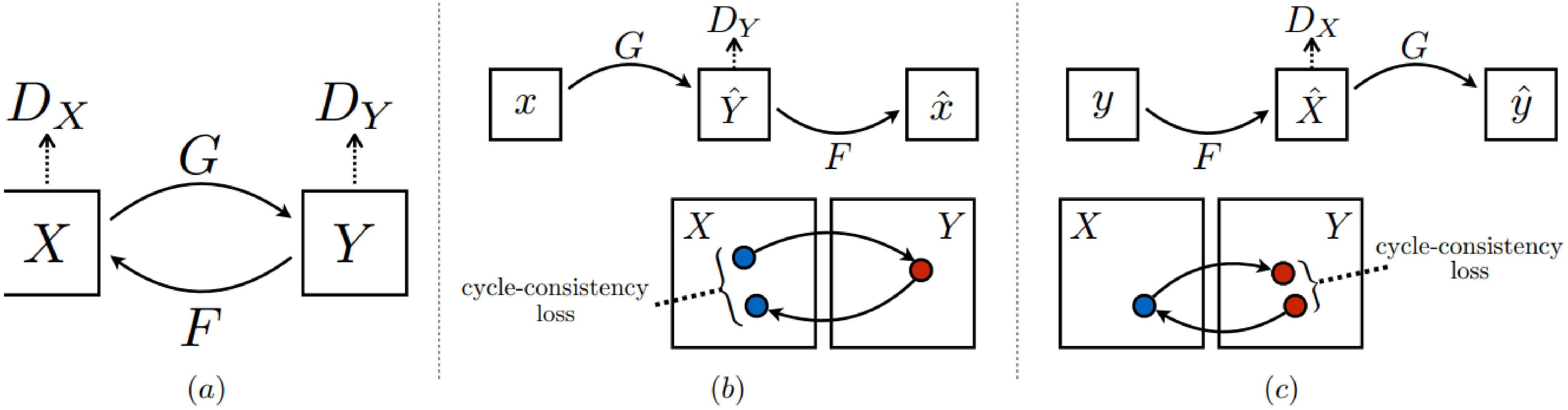




Cycle GAN

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

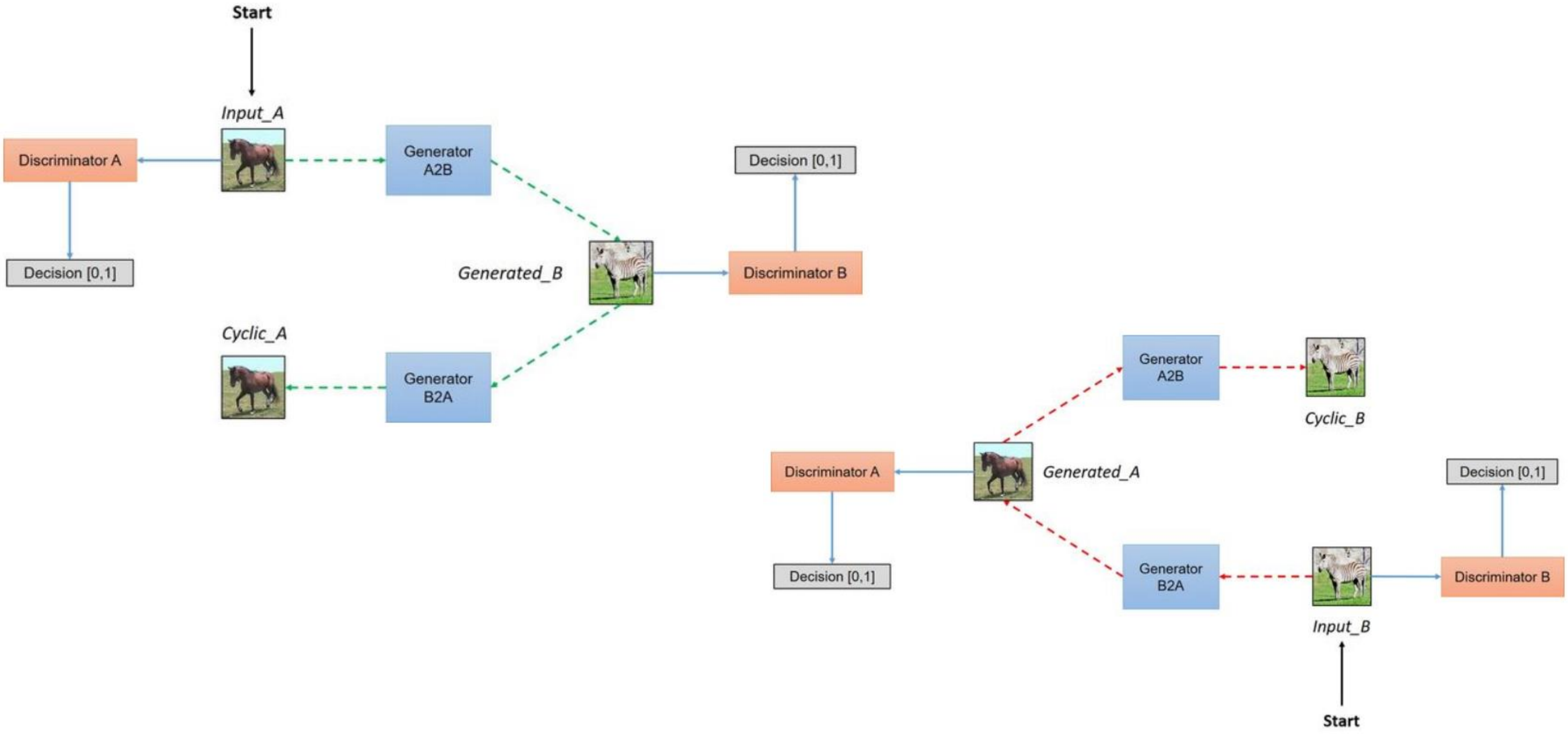
Jun-Yan Zhu* Taesung Park* Phillip Isola Alexei A. Efros
Berkeley AI Research (BAIR) laboratory, UC Berkeley



Cycle GAN

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

Jun-Yan Zhu* Taesung Park* Phillip Isola Alexei A. Efros
Berkeley AI Research (BAIR) laboratory, UC Berkeley



Cycle GAN

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

Jun-Yan Zhu* Taesung Park* Phillip Isola Alexei A. Efros
Berkeley AI Research (BAIR) laboratory, UC Berkeley

$$L_{\mathrm{GAN}}(F, D_Y, X, Y) = E_{y \sim p_{\mathrm{data}}(y)}[\log D_Y(y)] + E_{x \sim p_{\mathrm{data}}(x)}[\log(1 - D_Y(F(x)))]$$

Original Lost function of GANs

$$\begin{aligned} L_{cyc}(G_{A2B}, G_{B2A}, A, B) &= \mathbb{E}_{x \sim A}[\|G_{B2A}(G_{A2B}(x)) - x\|_1] \\ &+ \mathbb{E}_{y \sim B}[\|G_{A2B}(G_{B2A}(y)) - y\|_1] \end{aligned}$$

$$G_{B2A}(G_{A2B}(x)) \simeq x$$
$$G_{A2B}(G_{B2A}(y)) \simeq y$$

Consider “Consistency Loss”

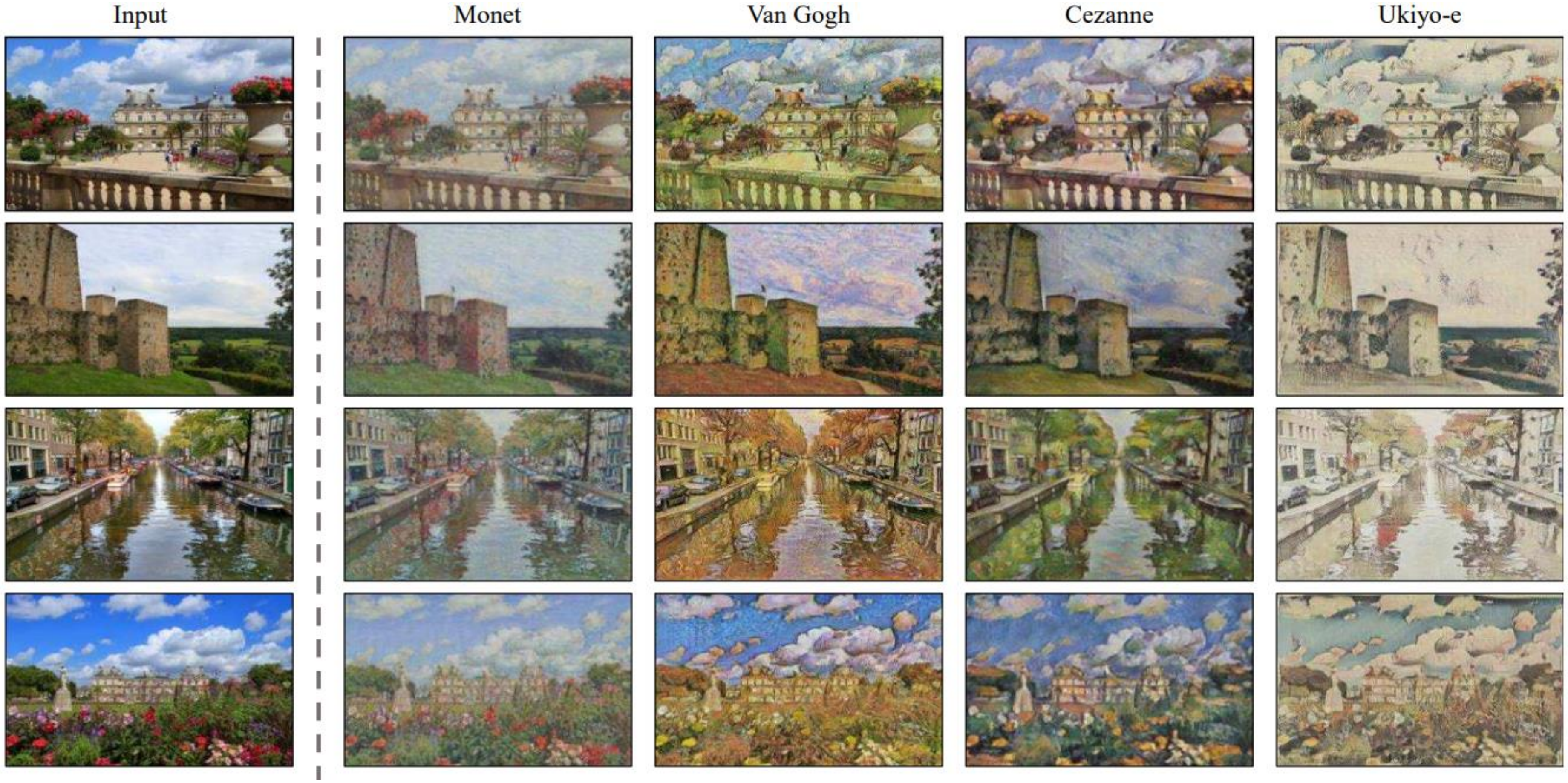
$$\begin{aligned} L(G_{A2B}, G_{B2A}, D_A, D_B) &= L_G(G_{A2B}, D_B, A, B) \\ &+ L_G(G_{B2A}, D_A, B, A) \\ &+ \lambda L_{cyc}(G_{A2B}, G_{B2A}, A, B) \end{aligned}$$

Lost function of Cycle-GANs

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

Jun-Yan Zhu* Taesung Park* Phillip Isola Alexei A. Efros
Berkeley AI Research (BAIR) laboratory, UC Berkeley

Result from the paper



Thank you!

- **More reference in :**

PyTorch

知乎

arXiv.org