

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

Ткачев Андрей, НИУ ВШЭ, 2019

План

- ▶ Задача Image-To-Image Translation, отличие paired и unpaired подходов
- ▶ Мотивация использовать unpaired и существующие unpaired методы
- ▶ CycleGAN особенности и принцип работы
- ▶ Используемые функции потерь
- ▶ Оптимизируемый функционал
- ▶ Архитектура генератора
- ▶ Архитектура дискриминатора
- ▶ Особенности обучения
- ▶ Сравнение
- ▶ Примеры и приложения

Задача Image-to-Image Translation

- ▶ Построить отображение:

- ▶ Такое, что:

$$G : X \rightarrow Y$$

$$\mathcal{P}_{G(X)} \sim \mathcal{P}_Y$$

Задача Image-to-Image Translation

Zebras Horses

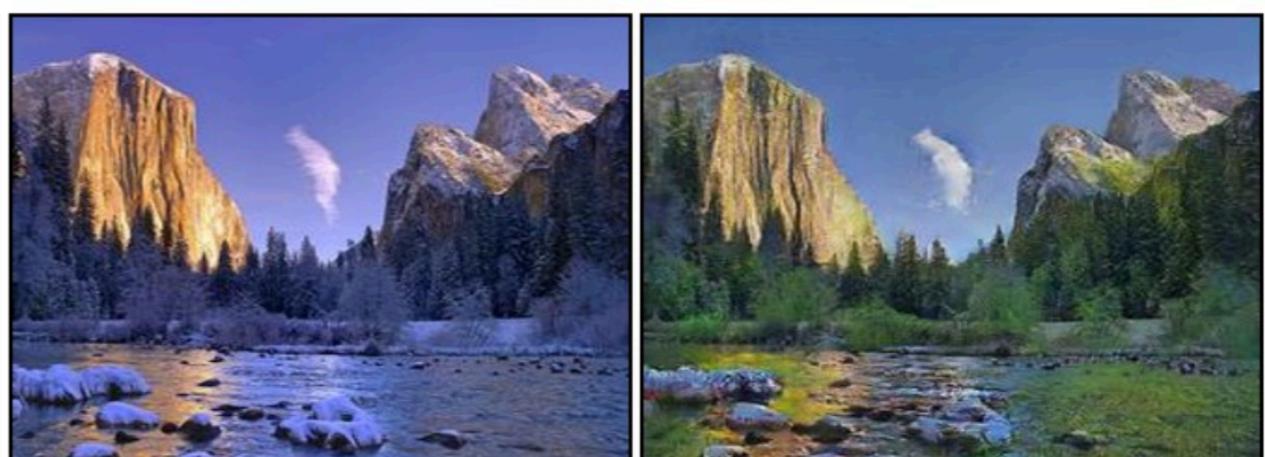


zebra → horse

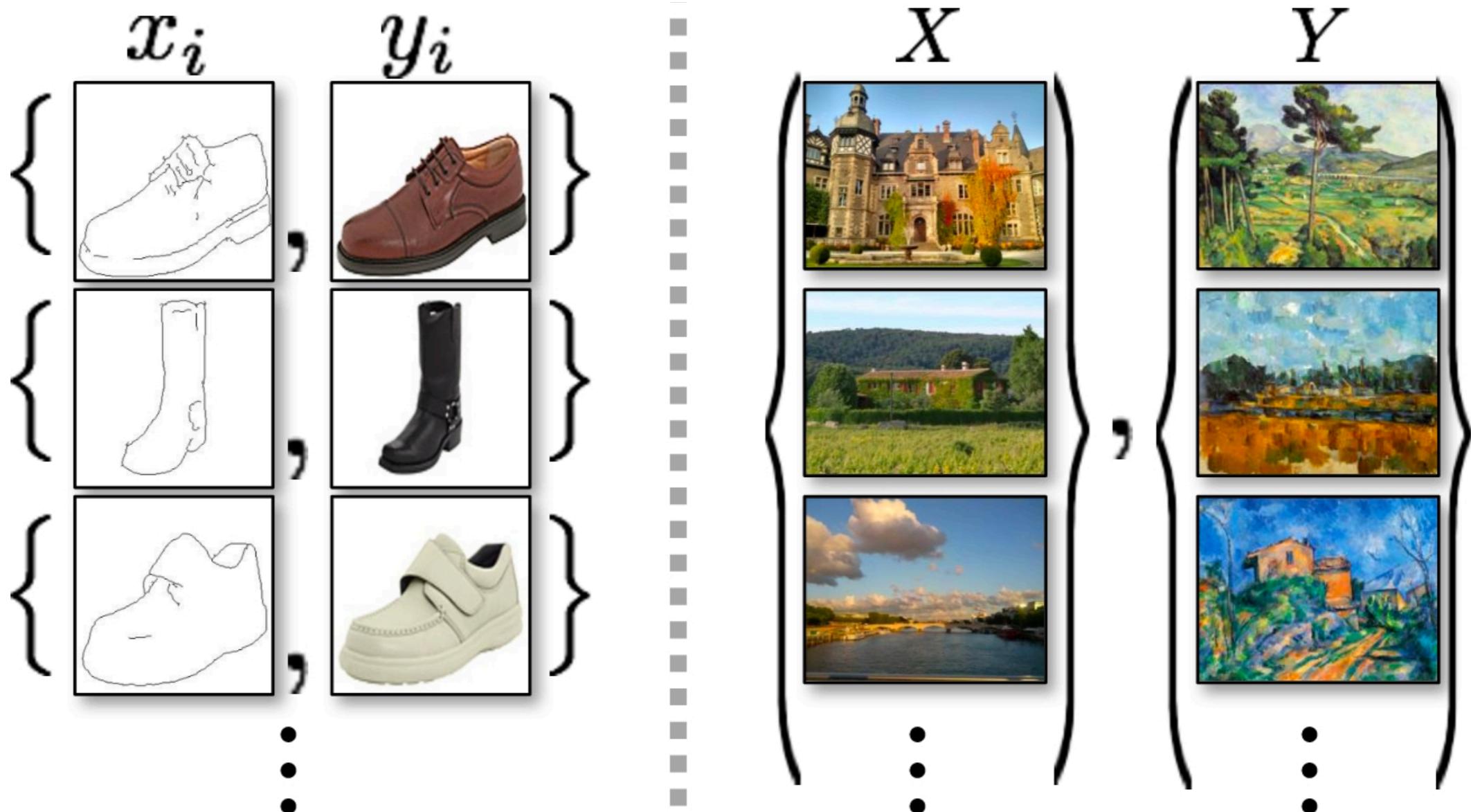
Summer Winter



summer → winter



Отличие paired и unpaired подходов



Мотивация использовать unpaired методы

- ▶ Не для всех задач просто получить размеченные датасеты
 - семантическа сегментация
 - стилизация изображений

Существующие unpaired методы

▶ CoGAN

- Несколько GAN-ами на первых слоях используются общие веса
- Приближает совместное распределение

▶ SimGAN

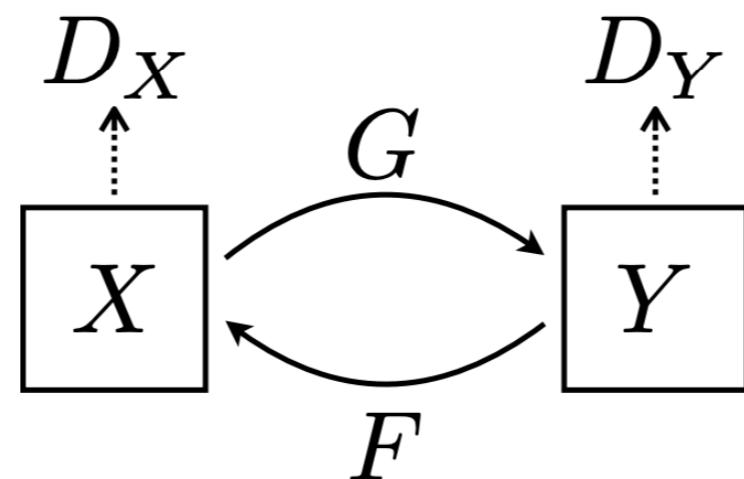
- Состязательная сеть с ограничениями на близость исходных данных и результата относительно в некотором определенном метрическом пространстве

Особенности CycleGAN

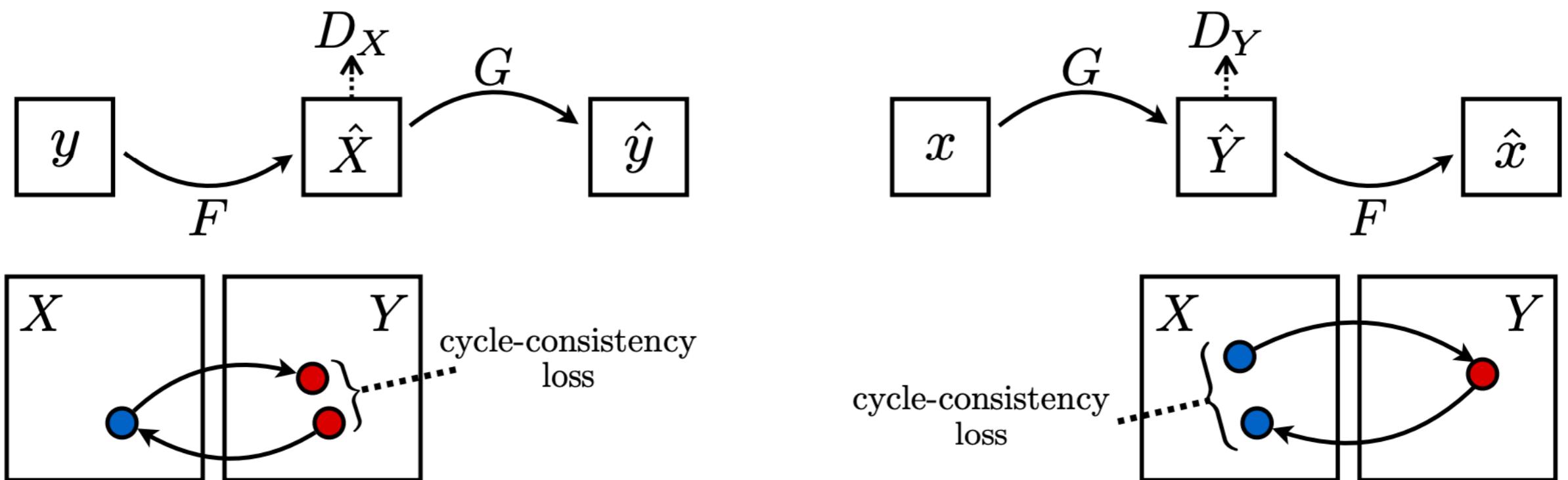
- ▶ Не зависит от природы данных
- ▶ Не требует предобученных функций близости изображений
- ▶ Обучается на не сопоставленных наборах изображений

Принцип работы CycleGAN

- ▶ Пусть X, Y — множества изображений (возможно разной мощности)
- ▶ Дискриминаторы множеств: D_X, D_Y
- ▶ Преобразования: $F : X \rightarrow Y, G : Y \rightarrow X$



Принцип работы CycleGAN



Adversarial Loss

- ▶ Соревновательная функция потерь:

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

- ▶ Оптимизируем:

$$\min_G \max_{D_Y} \mathcal{L}_{GAN}(G, D_Y, X, Y)$$

Cycle Consistency Loss

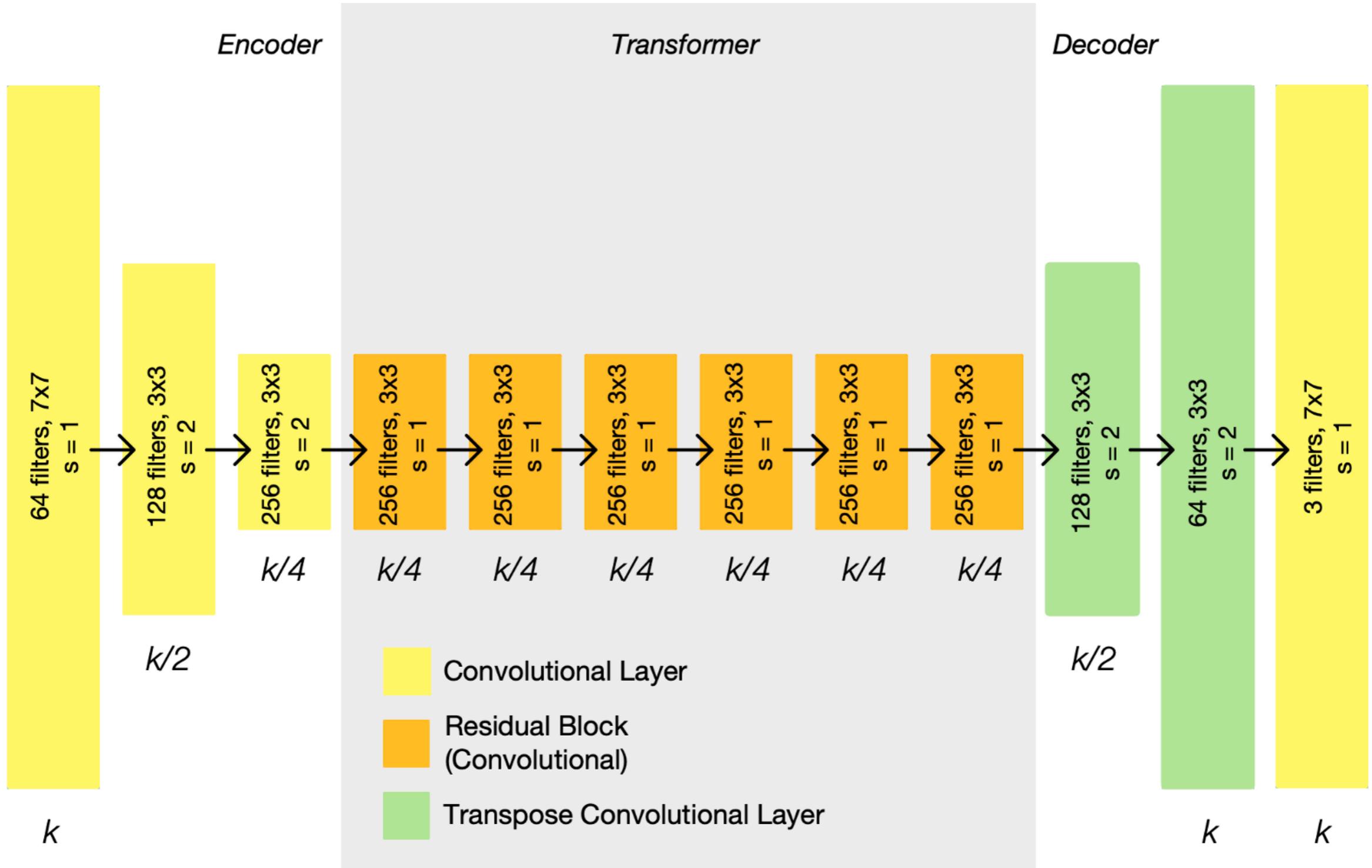
$$\begin{aligned}\mathcal{L}_{cyc}(G, F) = & \mathbb{E}_{x \sim p_{data(x)}} [\| F(G(x)) - x \|_1] \\ & + \mathbb{E}_{y \sim p_{data(y)}} [\| G(F(y)) - y \|_1]\end{aligned}$$

Оптимизируемый функционал

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

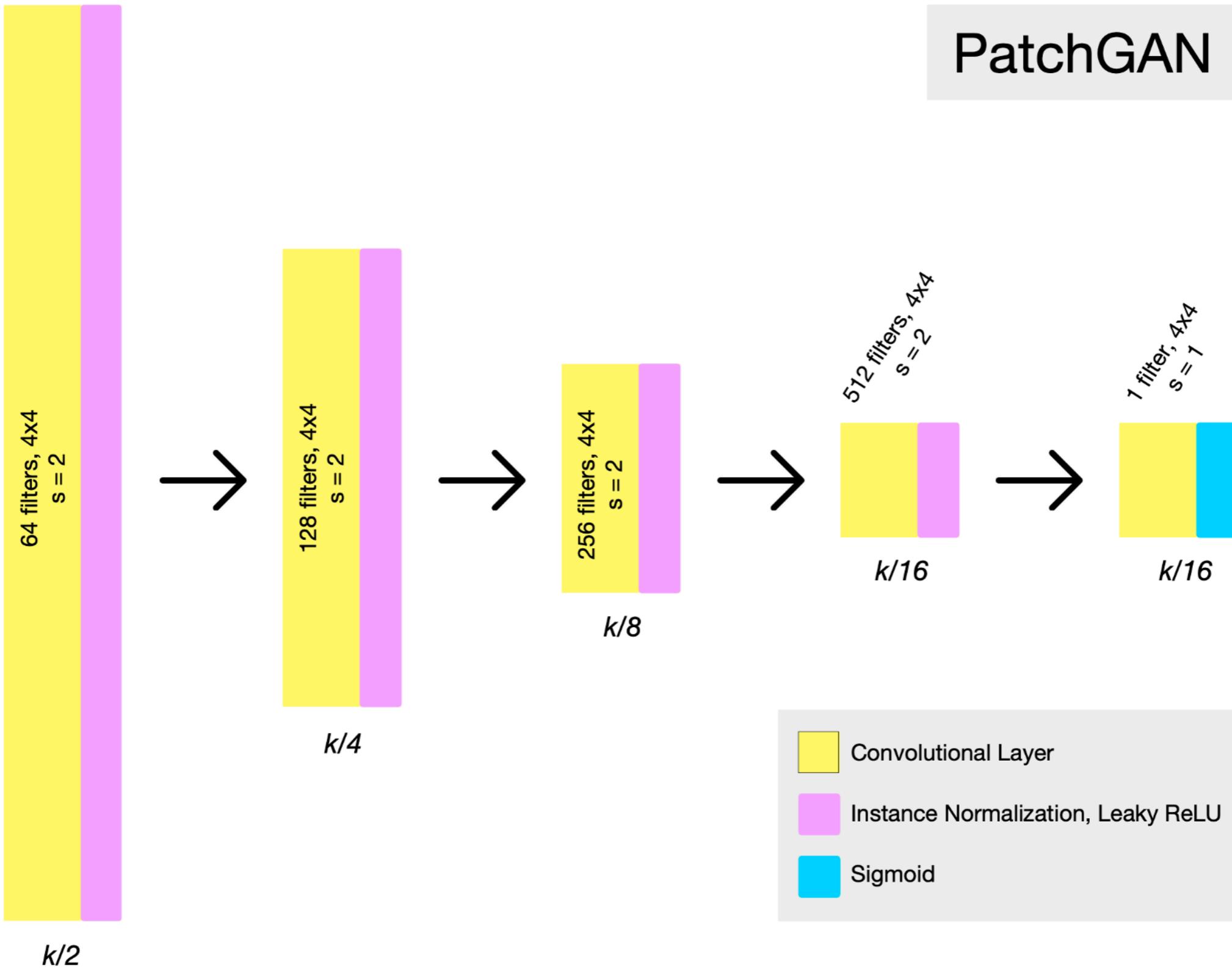
$$G^*, F^* = \arg \min_{G, F} \max_{D_Y, D_X} \mathcal{L}(G, F, D_Y, D_X)$$

Архитектура генератора



*InstanceNormalisation + ReLu после каждого слоя

Архитектура дескриминатора



Особенности обучения

- ▶ Обновляя веса дискриминатора используем историю из 50 сгенерированных изображений (вместо последнего), для которых мы точно знаем, что они фальшивые
 - Уменьшает осцилляции при обучении
- ▶ В adversarial loss вместо отрицательного логарифма правдоподобия используем least-squares loss
 - Более стабильная функция потерь

Сравнения

Loss	Map → Photo	Photo → Map
	% Turkers labeled <i>real</i>	% Turkers labeled <i>real</i>
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%

Table 1: AMT “real vs fake” test on maps↔aerial photos at 256×256 resolution.

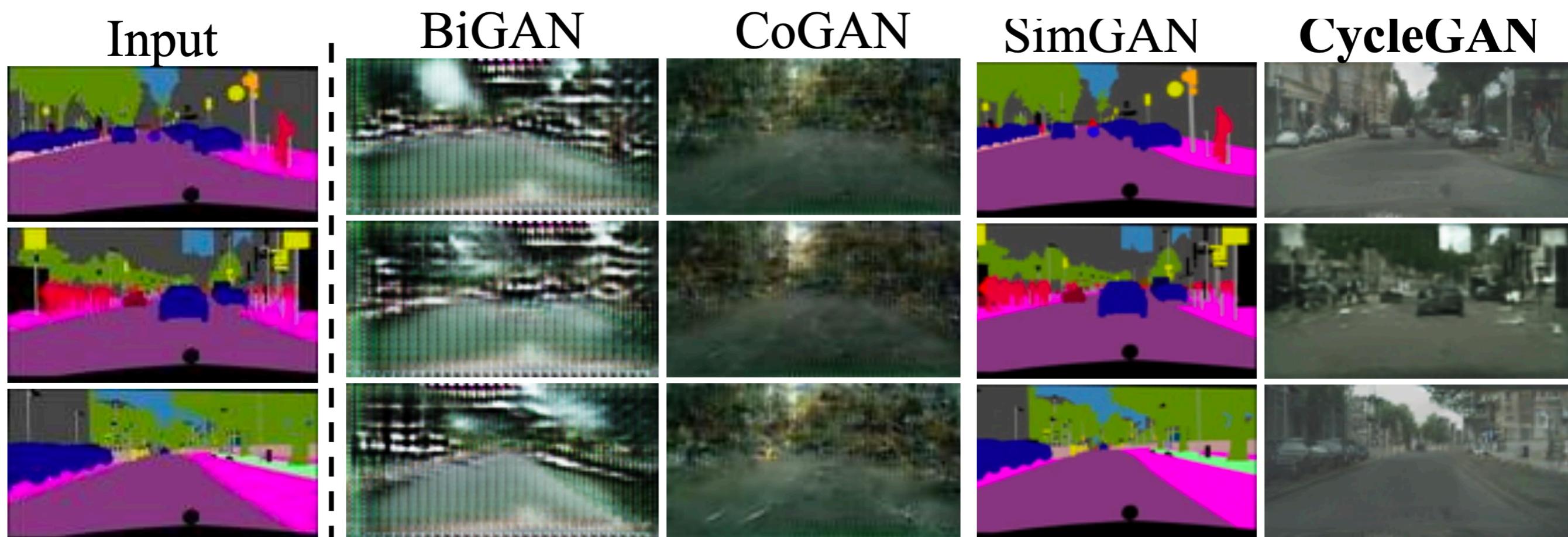
Сравнения

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

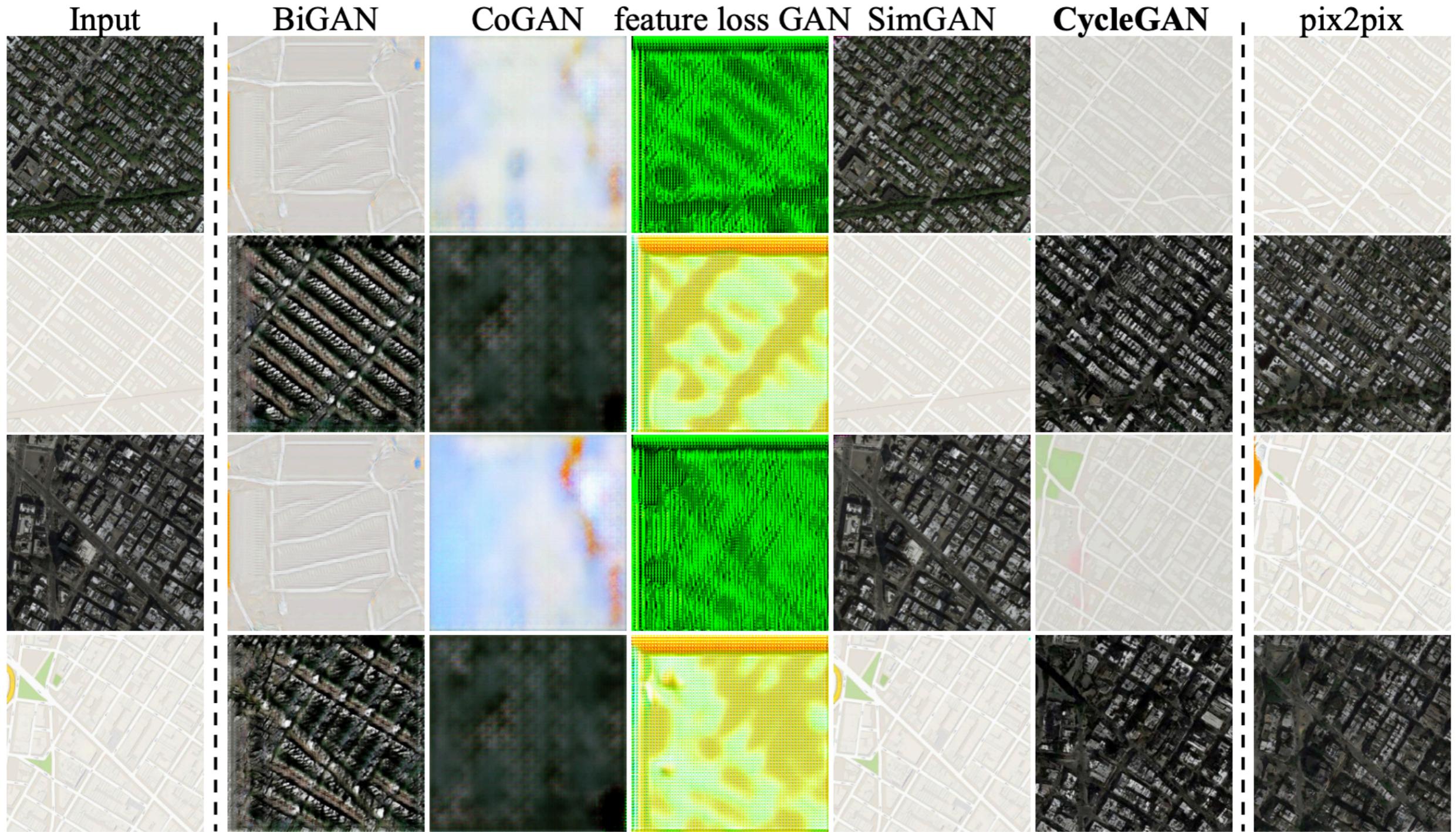
Table 2: FCN-scores for different methods, evaluated on Cityscapes labels→photo.

*fully-convolutional network scores

Сравнения



Сравнения

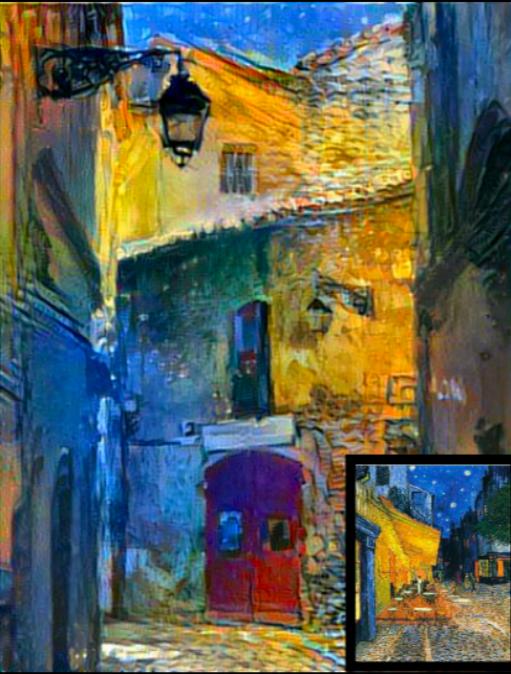


Сравнения

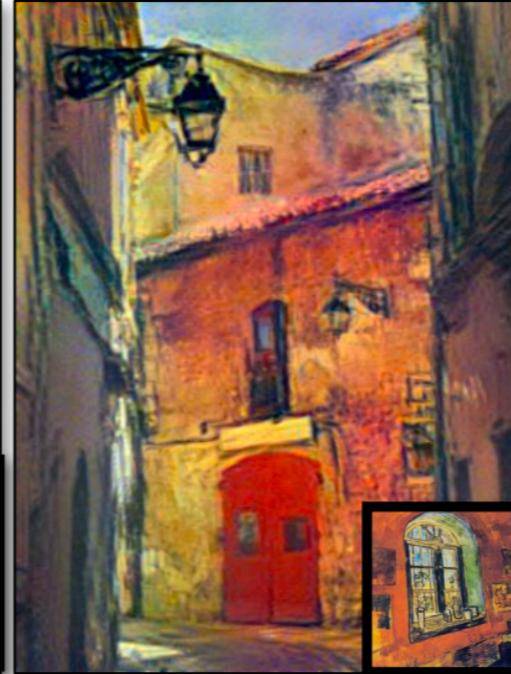
Input



Gatys et al. (image I)



Gatys et al. (image II)



Gatys et al. (collection)



CycleGAN

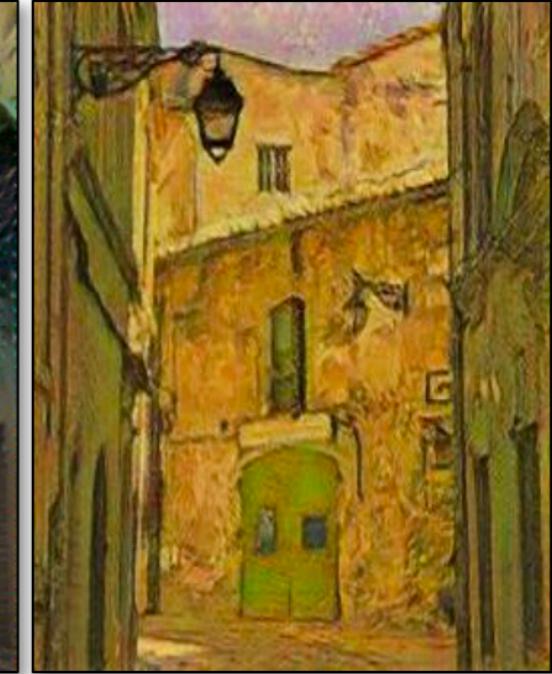


Photo → Van Gogh



Photo → Ukiyo-e



Photo → Cezanne

Примеры и приложения

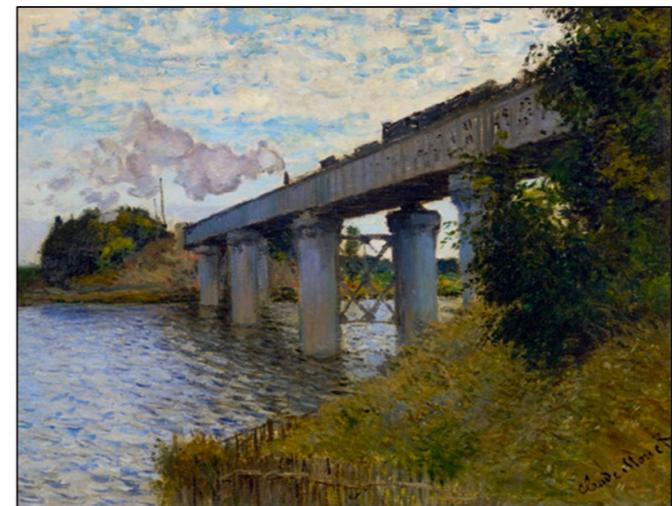
Input



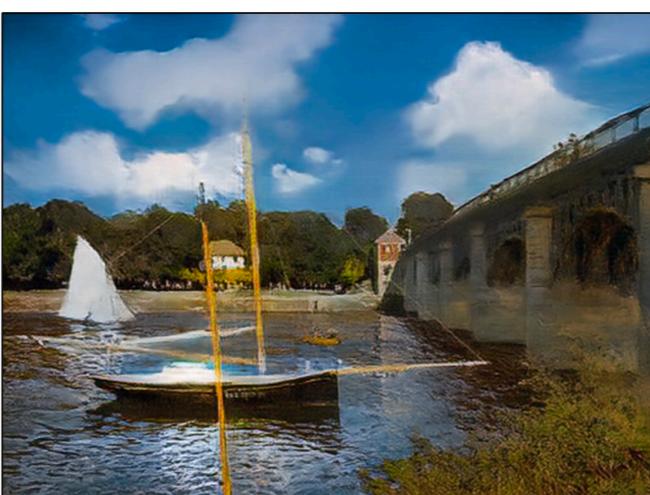
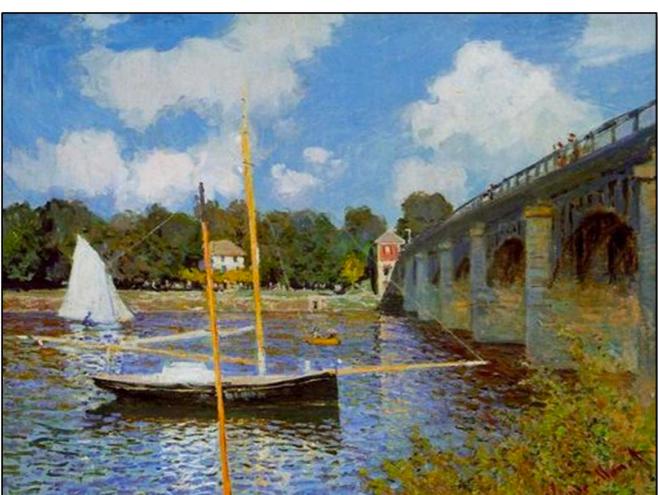
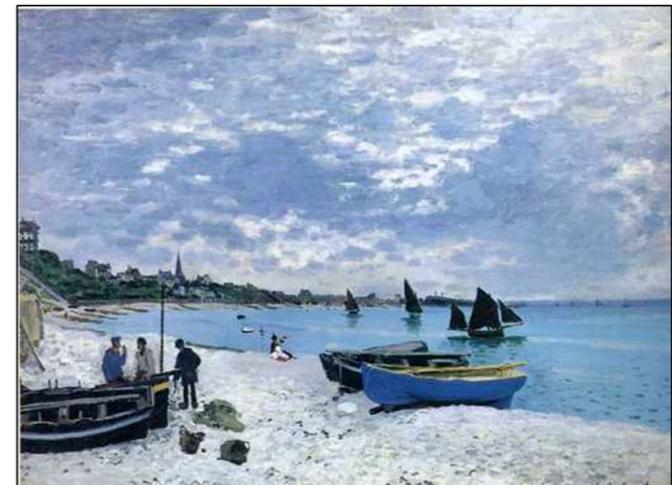
Output



Input



Output



Примеры и приложения

Input



Monet



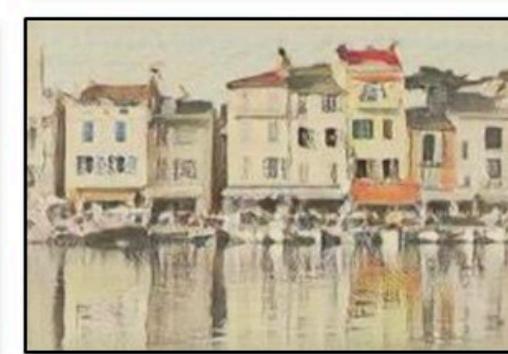
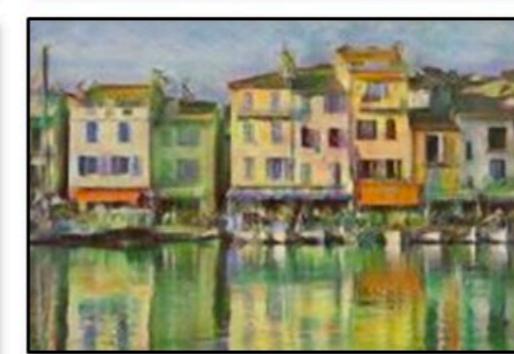
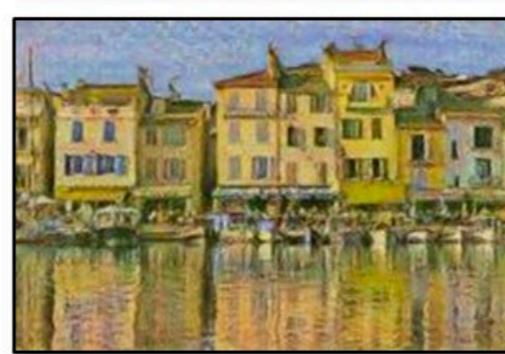
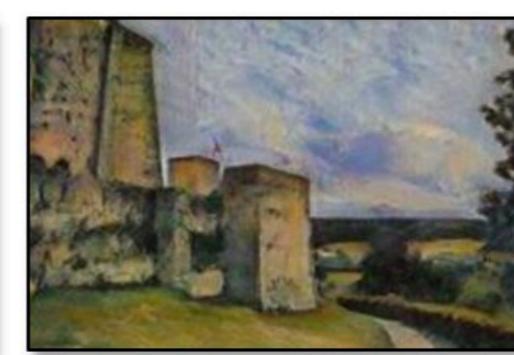
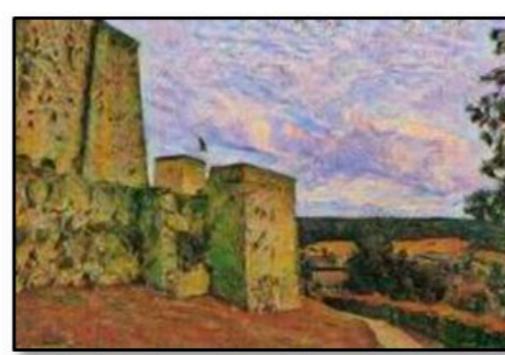
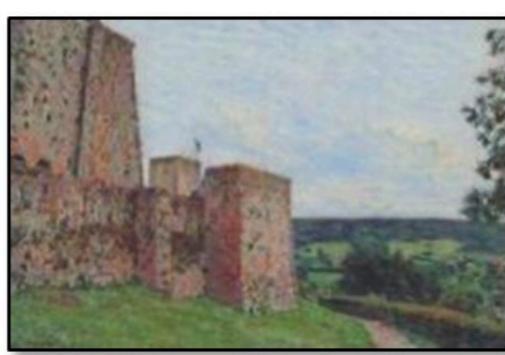
Van Gogh



Cezanne



Ukiyo-e



Примеры и приложения

Input



Output



Примеры и приложения



Примеры и приложения



Примеры неудач



ИТОГИ

- ▶ CycleGAN — метод unpaired Image-To-Image Translation
- ▶ Превосходит предшественников/конкурентов SimGAN, CoGAN
- ▶ По прежнему сильно отстает от paired методов вроде pix2pix

Материалы

- ▶ CycleGAN <https://arxiv.org/abs/1703.10593>
- ▶ Изображения в докладе <https://junyanz.github.io/CycleGAN/>