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

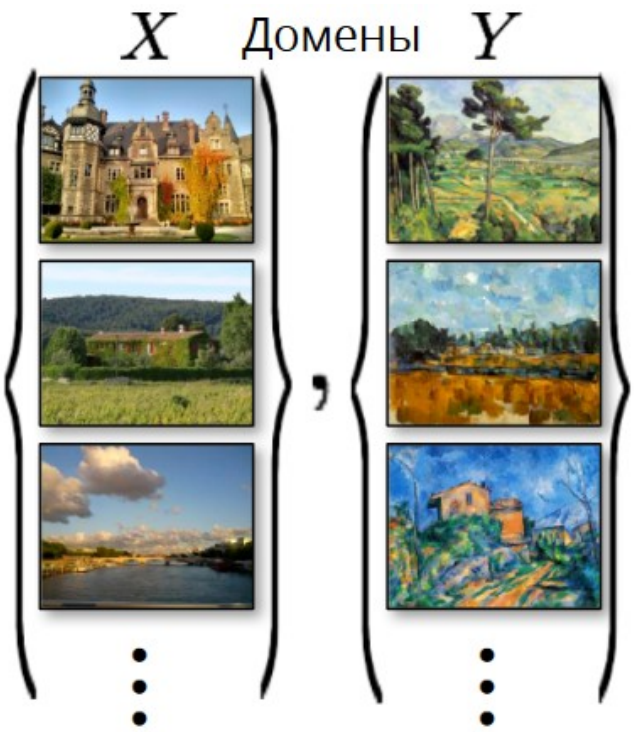
Image to image translation

Image-to-image translation - это класс задач где целью является поиск соответствия между входным и выходным изображением с использованием обучающего набора пар изображений.

Однако, не для всех задач существует набор из пар изображений, или же получение этих пар слишком затратно

В некоторых случаях желаемый результат даже нельзя чётко определить и сформулировать

$$\{x_i, y_i\}_{i=1}^{N}$$



$$\{x_i\}_{i=1}^{N} (x_i \in X)$$

$$\{y_i\}_{j=1}^{M} (y_j \in Y)$$

Ещё раз вспомним как работает GAN

Генератор *G* создаёт новые объекты из шума

Дискриминатор *D* отличает настоящие объекты от сгенерированных

Противопоставляем две модели друг другу и решаем следующую задачу:

$$\min_G \max_D L(D,G) = \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

$$= \mathbb{E}_{x\sim p_r(x)}[\log \underbrace{D(x)}] + \mathbb{E}_{x\sim p_g(x)}[\log \underbrace{(1-D(x)}]$$

Вероятность на настоящих объектах

Обратная вероятность на подделках

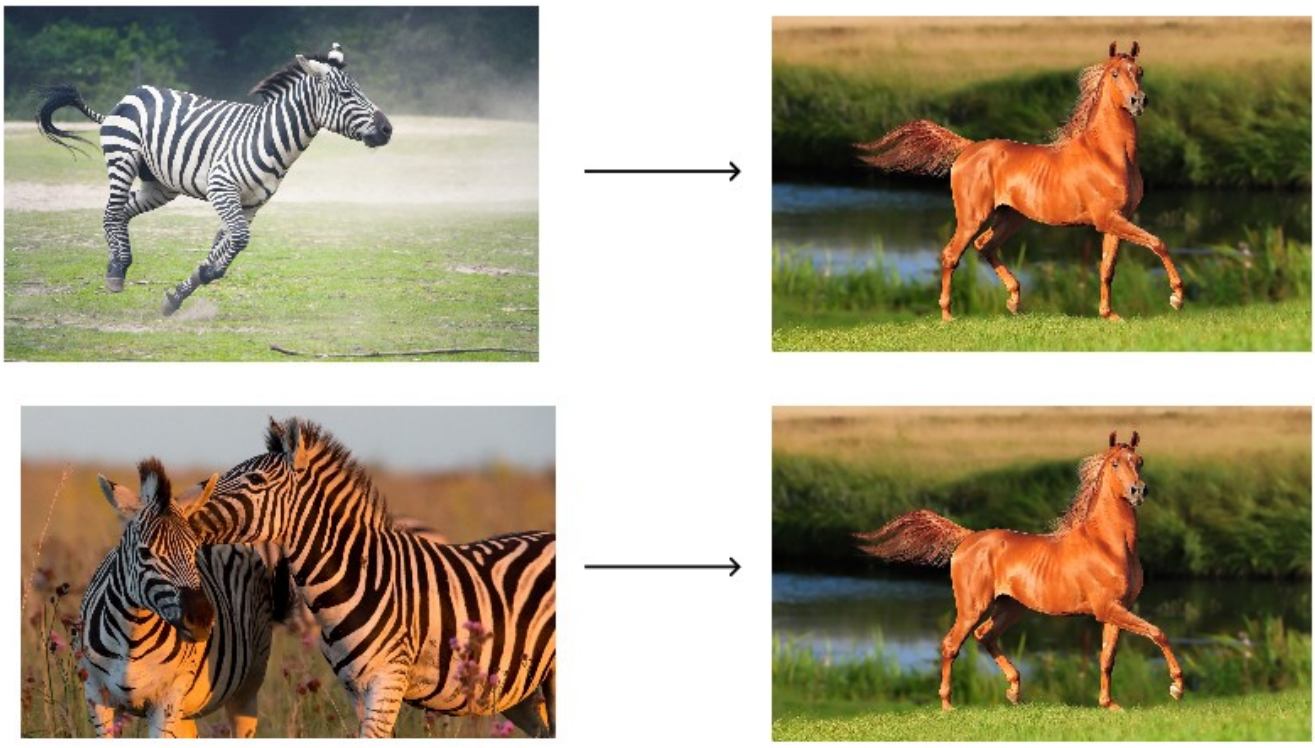
Попробуем использовать GAN в нашей задаче. Допустим, мы хотим обучить отображение вида

$$G : X \rightarrow Y$$

то есть, выход $G(x) = \hat{y}$ должен быть неотличим дискриминатором от изображений $y \in Y$

Однако такое отображение не гарантирует нам осмысленной связи между конкретными x и y

Более того, возможна и вполне вероятна ситуация, когда все входные изображения отображаются в единственное выходное изображение



Сделаем так, чтобы отображение имело свойство "cycle consistant", в том смысле, что

Лондон - столица Великобритании

London is the capital of Great Britain

Лондон - столица Великобритании

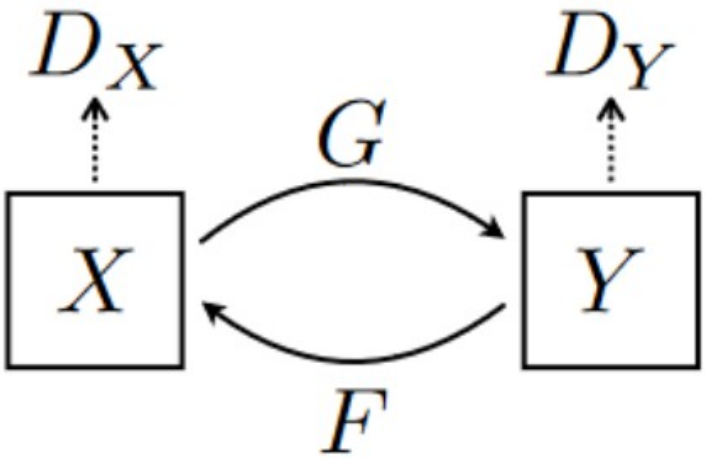
consistency - согласованность

Более формально, если $F : Y \to X$ и $G : X \to Y$

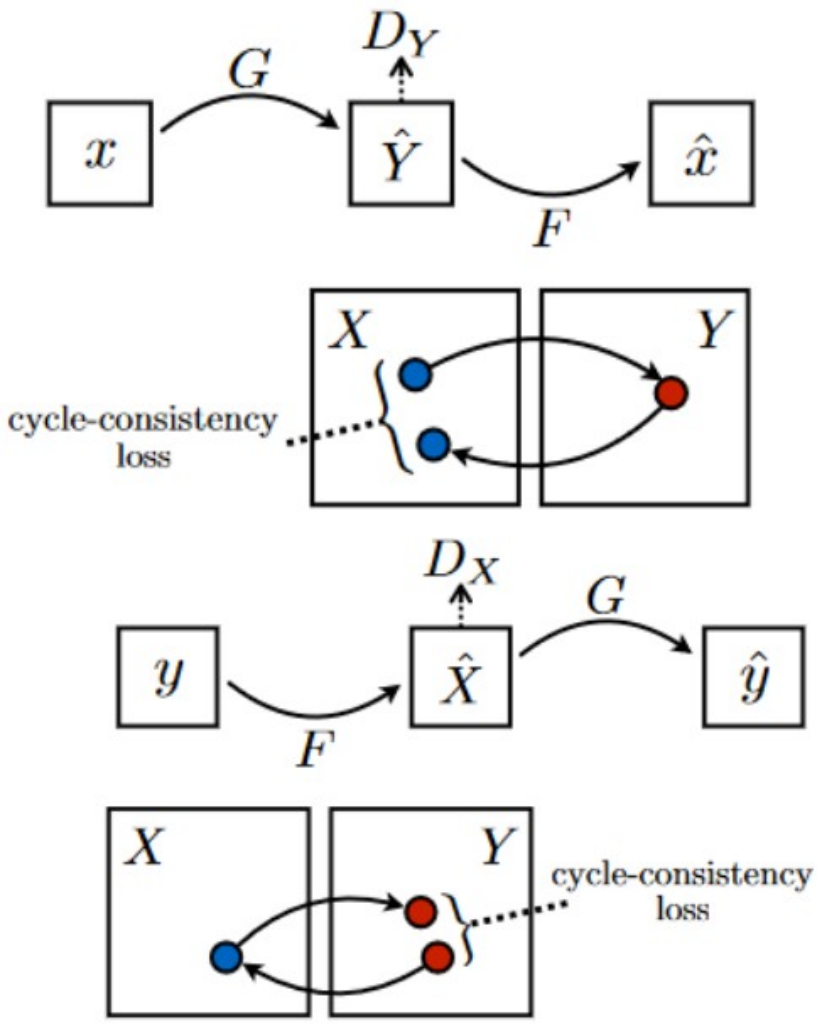
Тогда G и F являются обратными друг другу отображениями, и являются биекциями

Чтобы это получить, введём cycle consistency loss, которая поощряет

$$F(G(x)) \approx x \qquad G(F(y)) \approx y$$



Значит, имеем две функции отображения(генераторы) с соответствующими дискриминаторами Dx , Dy


D_Y
G
x
$\hat{Y}$
$\hat{x}$
F
X
Y
cycle-consistency
loss
D_X
G
y
$\hat{X}$
$\hat{y}$
F
X
Y
cycle-consistency
loss

Cycle consistency loss

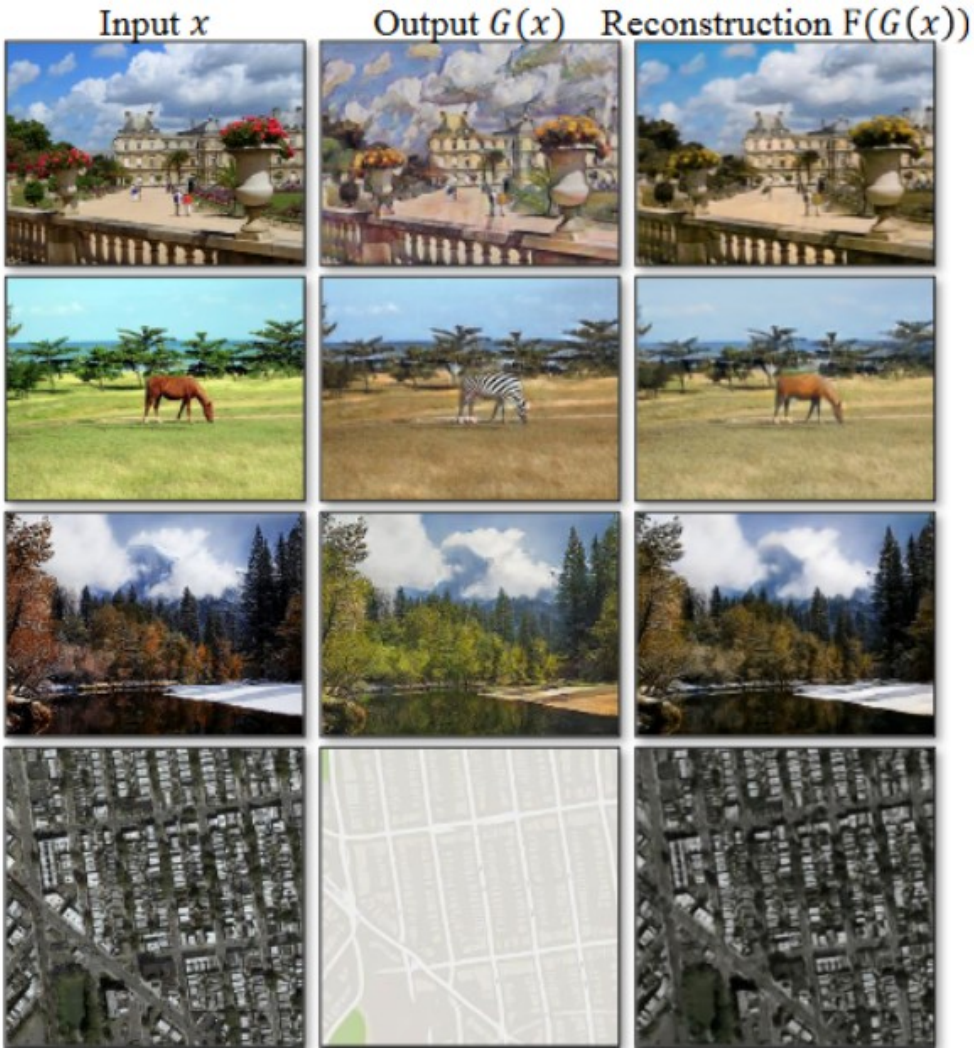
$$L_{cyc}(F, G) = \mathrm{E}_{x \sim p_{data}(x)}[\|F(G(x)) - x\|_1]$$

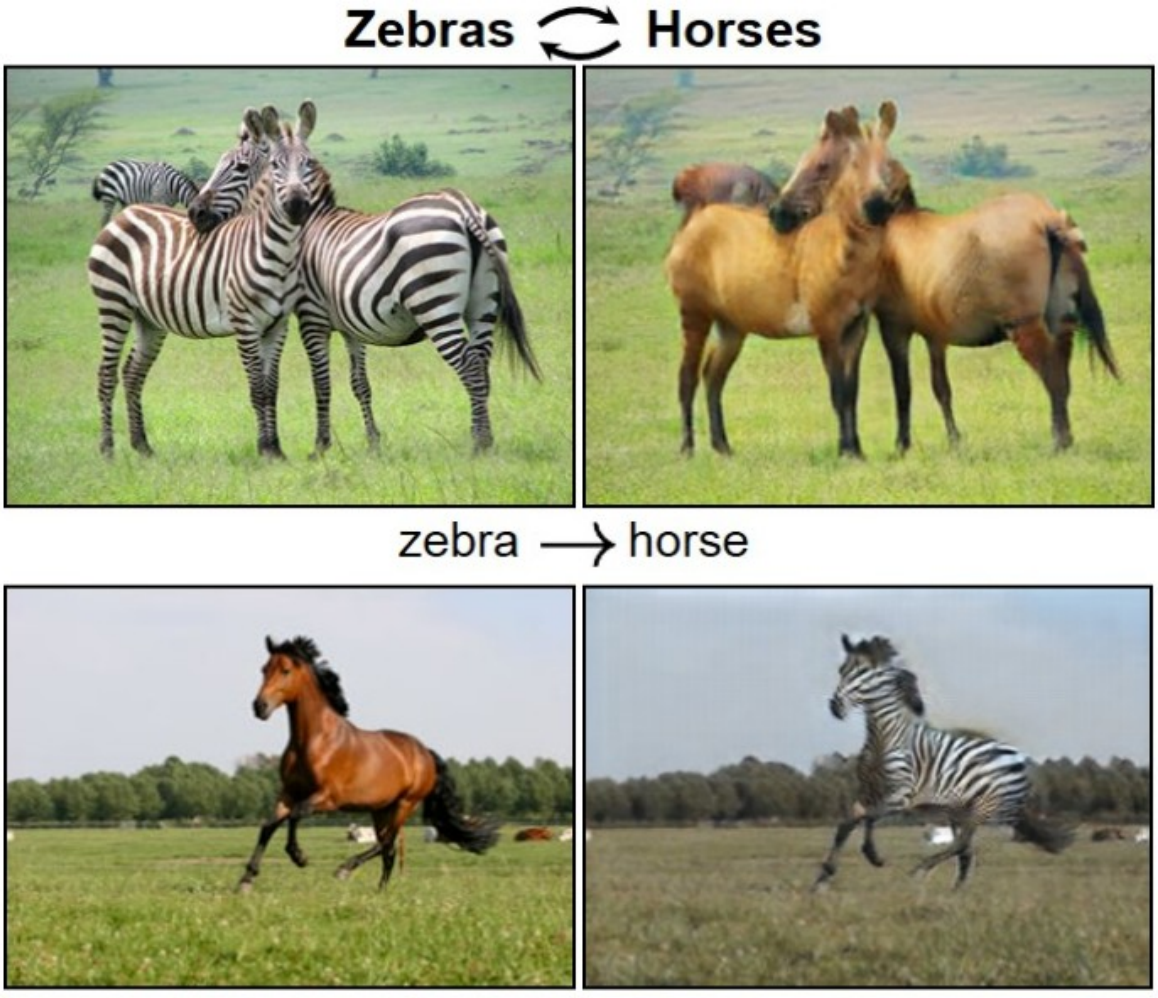
$$+\mathrm{E}_{y \sim p_{data}(y)}[\|G(F(y)) - y\|_1]$$

Итого, получаем следующую функцию потерь и задачу:

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

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

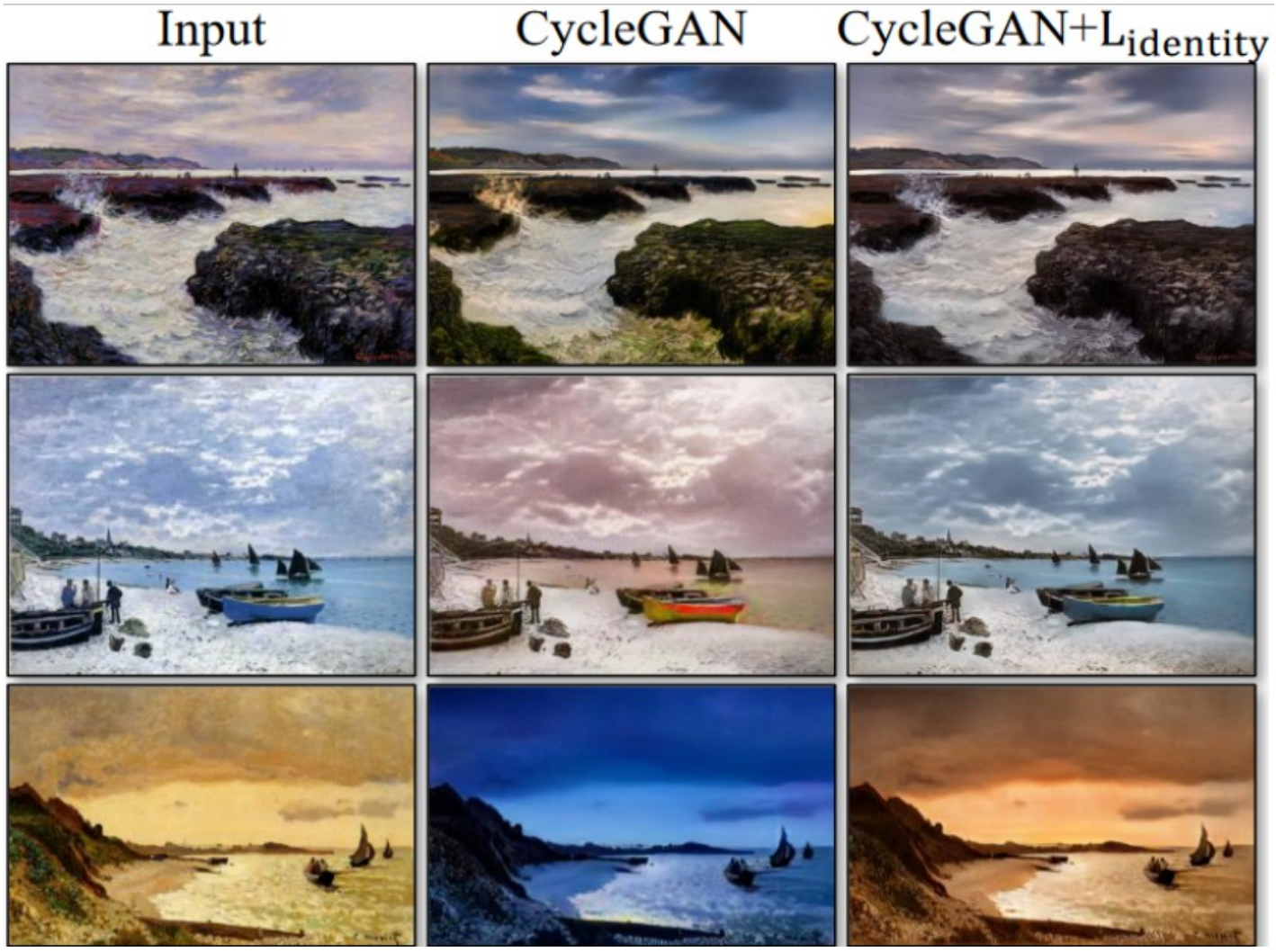

Input x
Output $G(x)$
Reconstruction F$(G(x))$

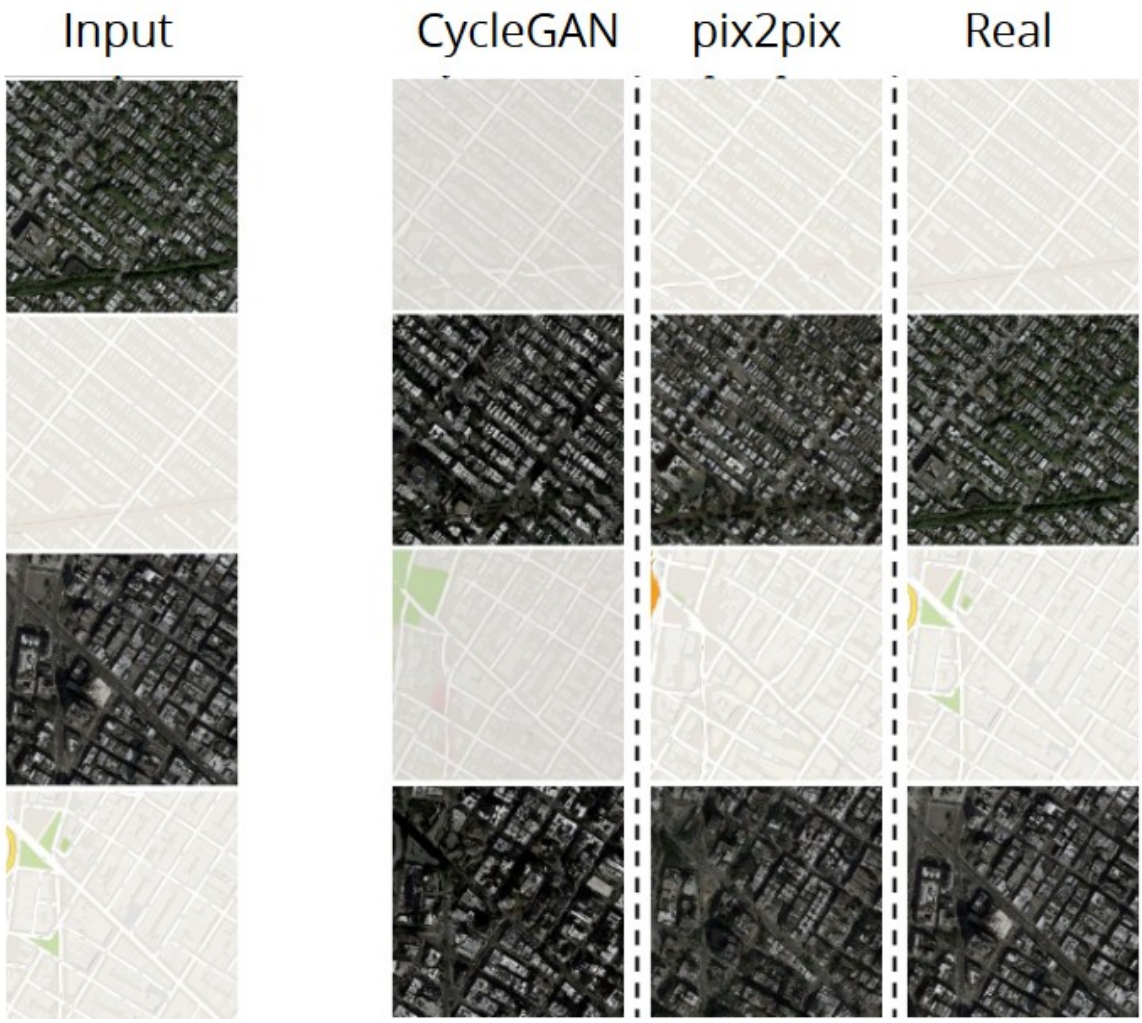

Zebras ⇄ Horses
zebra → horse
horse → zebra

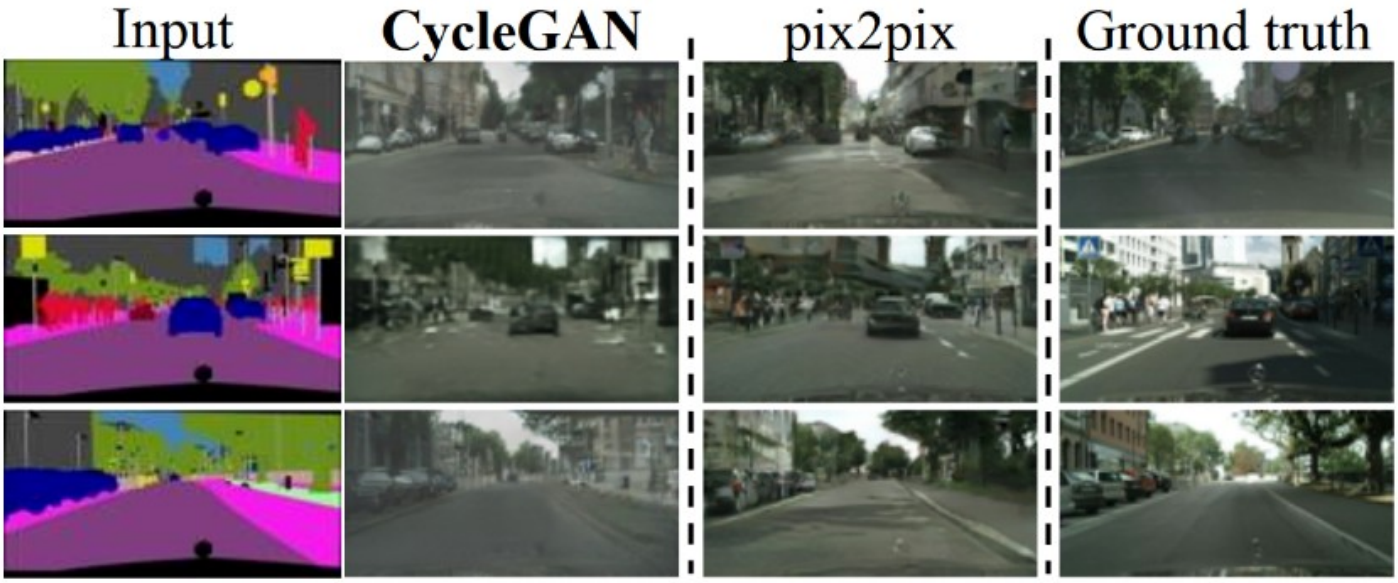
Но и это ещё не всё!

Боремся с произвольной цветовой гаммой

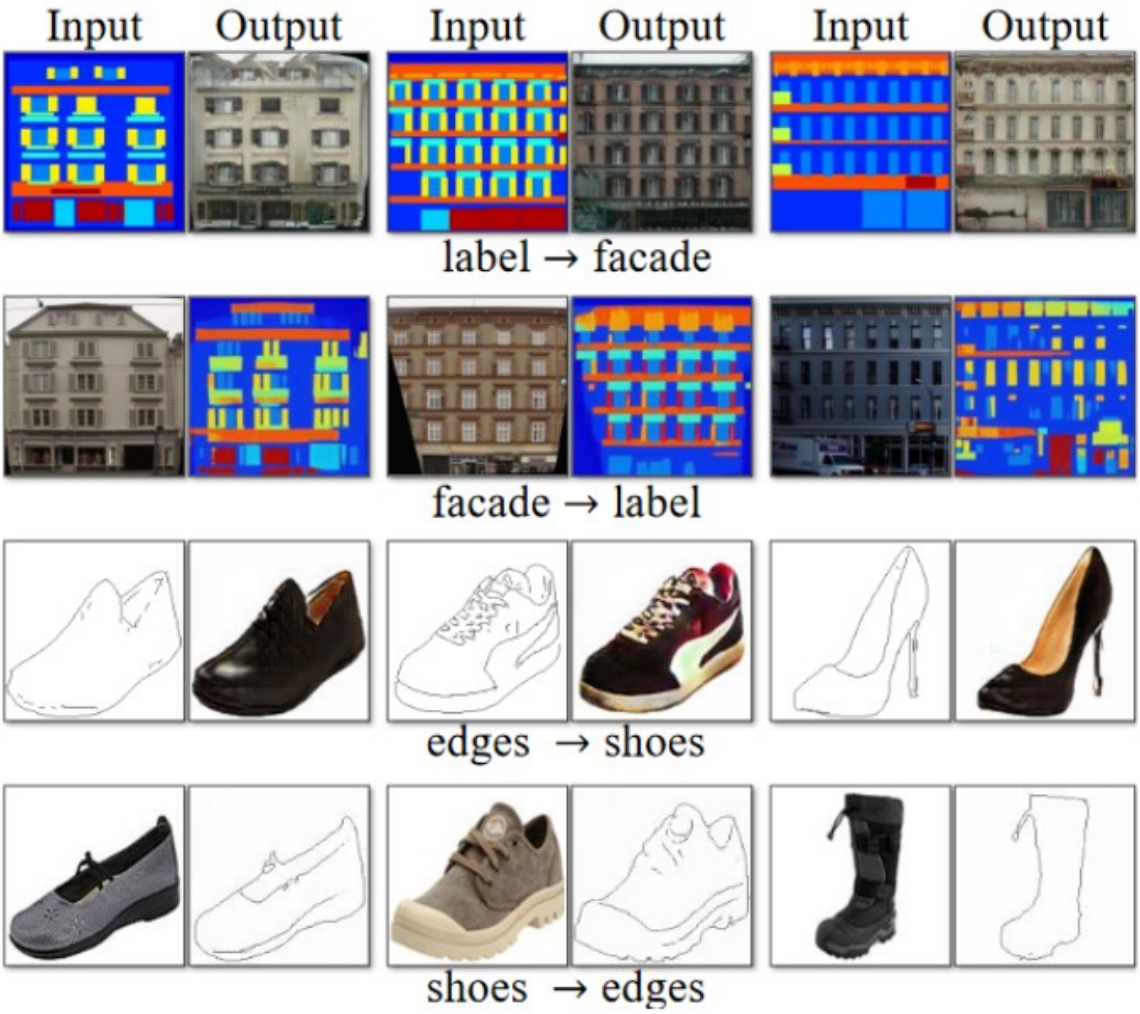
$$L_{identity}(G,F) = \mathrm{E}_{y\sim p_{data}(y)}[\|G(y)-y\|_1] + \mathrm{E}_{x\sim p_{data}(x)}[\|F(x)-x\|_1]$$


Input
CycleGAN
CycleGAN+L identity


Input
CycleGAN
pix2pix
Real



На этой задаче у BiGan, CoGan, SimGan было такое плохое качество, что они постеснялись появляться в этой презентации


Input
Output
Input
Output
Input
Output
label → facade
facade → label
edges → shoes
shoes → edges

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%**

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

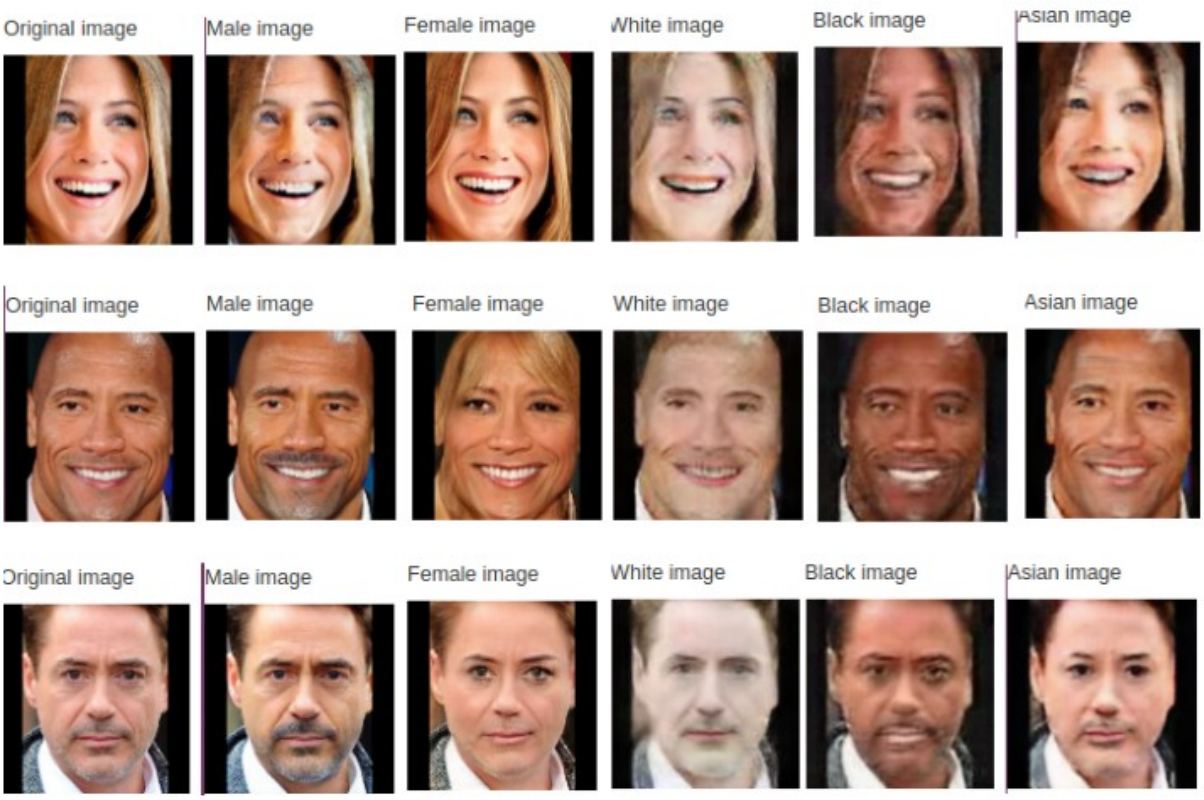
Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [32]	0.40	0.10	0.06
BiGAN/ALI [9, 7]	0.19	0.06	0.02
SimGAN [46]	0.20	0.10	0.04
Feature loss + GAN	0.06	0.04	0.01
CycleGAN (ours)	**0.52**	**0.17**	**0.11**
pix2pix [22]	0.71	0.25	0.18

Table 2: 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

Table 3: Classification performance of photo→labels for different methods on cityscapes.

Смена пола, расы, возраста и т.д.



Ещё более женственны

Внимание! Очень сложный кейс!

Архитектуры сетей (генератор)

c7s1-k - 7×7 Convolution-InstanceNorm-ReLU layer with k filters and stride 1

dk - 3×3 Convolution-InstanceNorm-ReLU layer with k filters and stride2

Rk - residual block that contains two 3×3 con-volutional layers with the same number of filters on bothlayer

uk - a 3×3 fractional-strided-Convolution-InstanceNorm-ReLU layer with k filters and stride 1/2

1) c7s1-64, d128, d256, R256, R256, R256, R256, R256, R256, u128, u64, c7s1-3

2) c7s1-64, d128, d256, R256, R256, R256, R256, R256, R256, R256, R256, R256, R256, u128, u64, c7s1-3

Архитектуры сетей (дискриминатор)

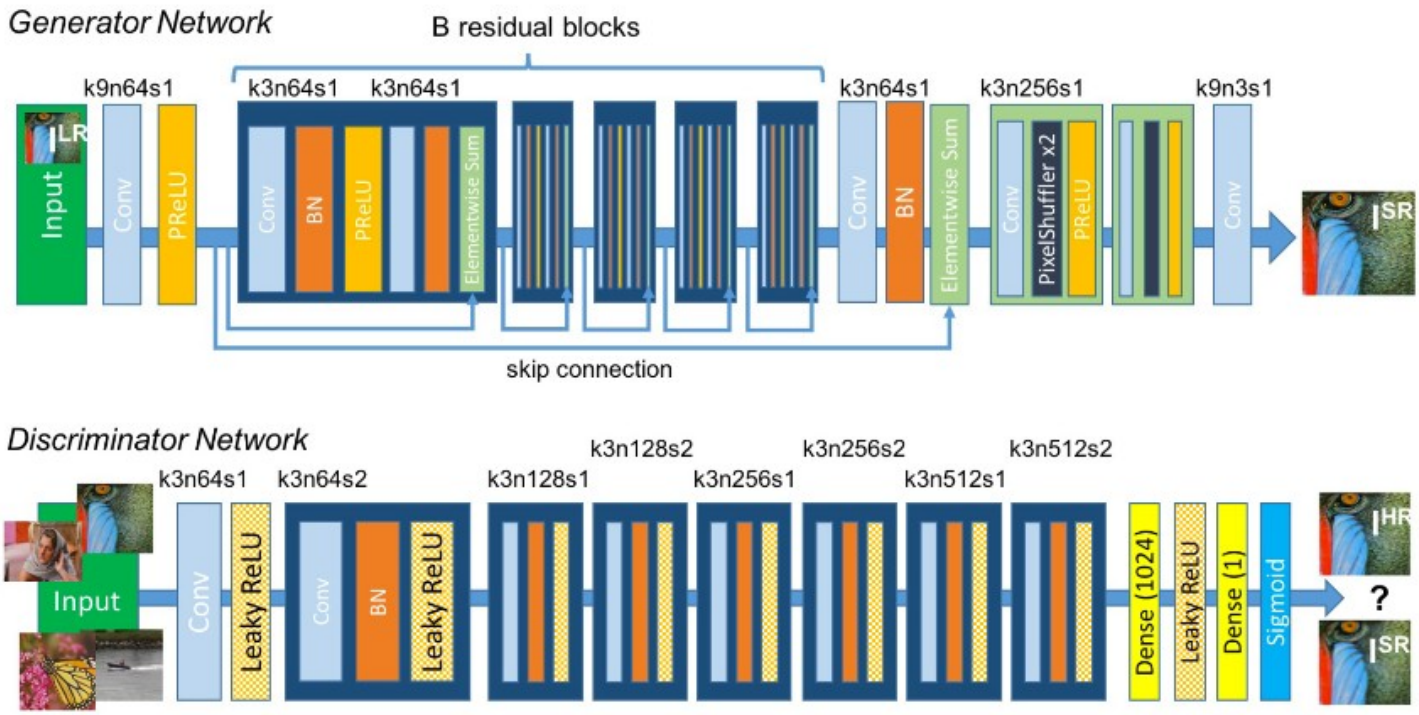
Ck - a 4×4 Convolution-InstanceNorm-LeakyReLU layer with k filters and stride2

InstanceNorm не применяется для C64

Архитектура: C64-C128-C256-C512

100 эпох с learning rate 0.0002, затем его линейно уменьшать до 0 в течение следующих 100 эпох

Архитектуры сетей



Ещё немного примеров

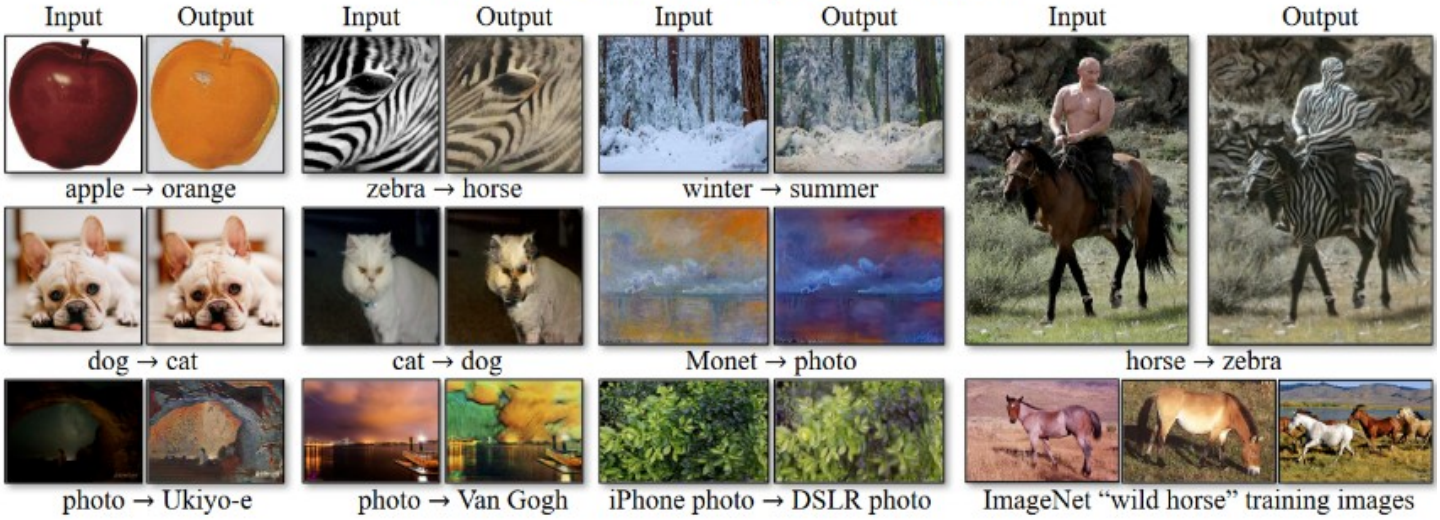
Input winter image

AI-generated summer image

Input sunny image

AI-generated rainy image

Известные проблемы



Здесь видно, что

1) Плохо меняет форму объектов
2) Вносит лишь небольшие изменения
3) В датасете нет фотографий лошадей со всадниками

Интересно, что

Во всех моделях использовалось всего 939 фотографий лошадей, 1177 фотографий зебр, 1074 картин Моне, 401 картина Ван Гога (поздние работы с наиболее узнаваемым стилем), 6853 реальных фотографий, 1096 изображений и Google maps.

Иными словами, датасеты не столь большие, а нейросети не такие глубокие как могло бы показаться сначала.

Все материалы

1) https://arxiv.org/pdf/1703.10593 (основная статья)

2) https://habr.com/ru/company/ods/blog/340154/ (лица, архитектура)

3) https://blogs.nvidia.com/blog/2017/12/03/nvidia-research-nips/ (пример от nvidia)

4) https://github.com/soumith/ganhacks (советы по обучению GAN-ов)

5) youtube: Turning a horse video into a zebra video (by CycleGAN)

(просто интересно посмотреть)