

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

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

Faculty of Computer Science
Higher School of Economics

Feb. 02, 2018

О чём сегодняшняя презентация

- 1 В целом о методе
- 2 Подробнее о составных частях
- 3 Формулировка
- 4 Оценки качества
- 5 Сравнение с другими методами
- 6 Основные сложности
- 7 Где применять

В целом о методе

- Цель — возможность «переводить» изображения из одного множества в другое, при этом не имея попарного перевода;
- Используем такие отображения $G : X \rightarrow Y$ и $F : Y \rightarrow X$, что $G(F(y)) \approx y$ и $F(G(x)) \approx x$;
- Логично использовать там, где нельзя или сложно получить попарные данные для обучения (например, трансформация объекта или смена сезона на фотографии);

В целом о методе



apple → orange



orange → apple



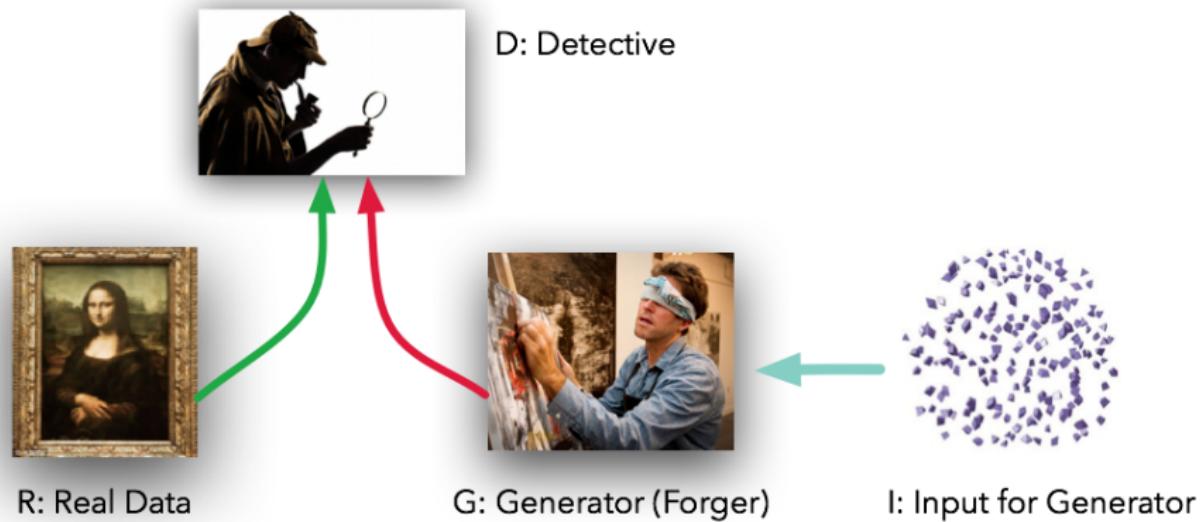
zebra → horse



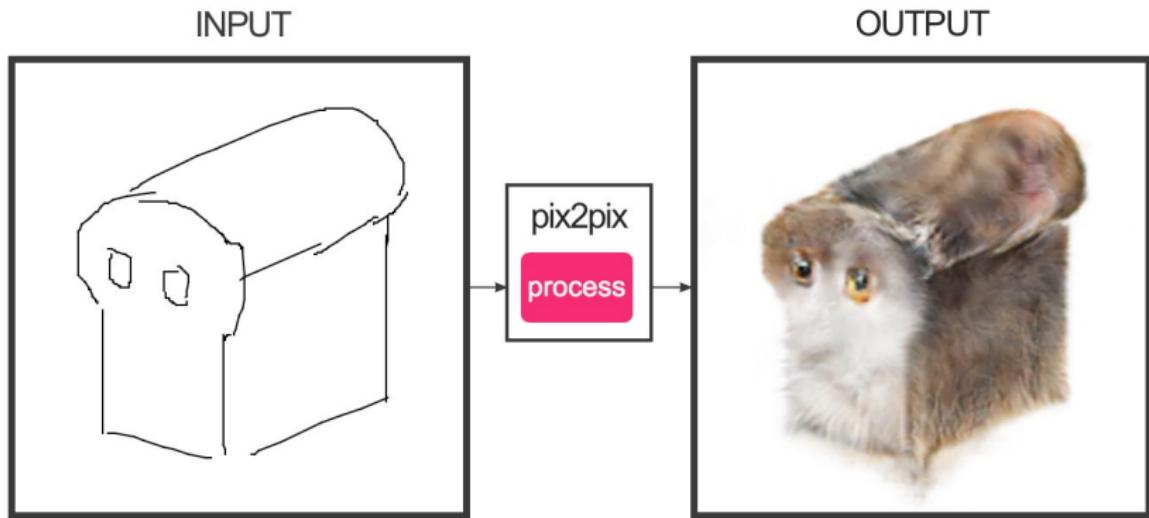
horse → zebra

GANs

Вкратце: две нейронные сети, одна генерирует данные, а другая
пытается отличить их от настоящих;



Unpaired Image-to-Image Translation



- Метод основывается на pix2pix;
- Отличие от предыдущего — непарный перевод;

Формулировка

- Два множества — X и Y , $x \sim p_{data}(x)$, $y \sim p_{data}(y)$;
- Два состязательных дискриминатора D_X, D_Y . D_X пытается отличить $\{x\}$ от $\{F(y)\}$, а D_Y — $\{y\}$ от $\{G(x)\}$;
- Будем использовать adversarial losses (чтобы распределение сгенерированных изображений приближалось к распределению настоящих) и cycle consistency losses (чтобы $G^{-1} \approx F$).

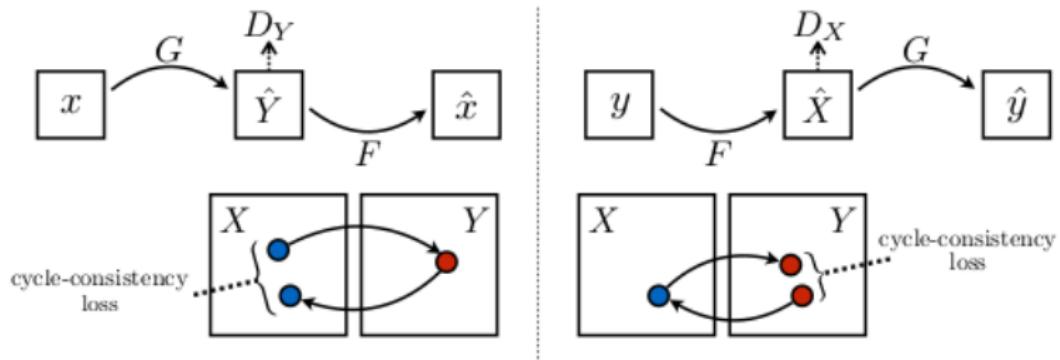
Adversarial Loss

- Формула:

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

- Аналогично для $\min_F \max_{D_X} \mathcal{L}_{GAN}(F, D_X, Y, X)$;

Cycle Consistency Loss



- Формула:

$$\mathcal{L}_{cyc}(G, F) =$$

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

Full Objective

- Формула:

$$\mathcal{L}(G, F, D_X, D_Y) =$$

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

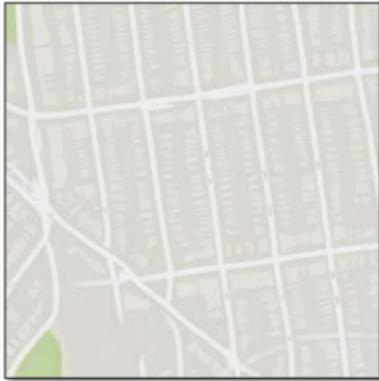
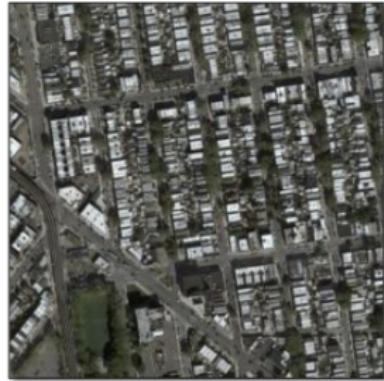
- Нужно решить:

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

- Дальше будет сравнение методов: просто GAN, GAN с adversarial loss, GAN с cycle consistency loss и т.д.

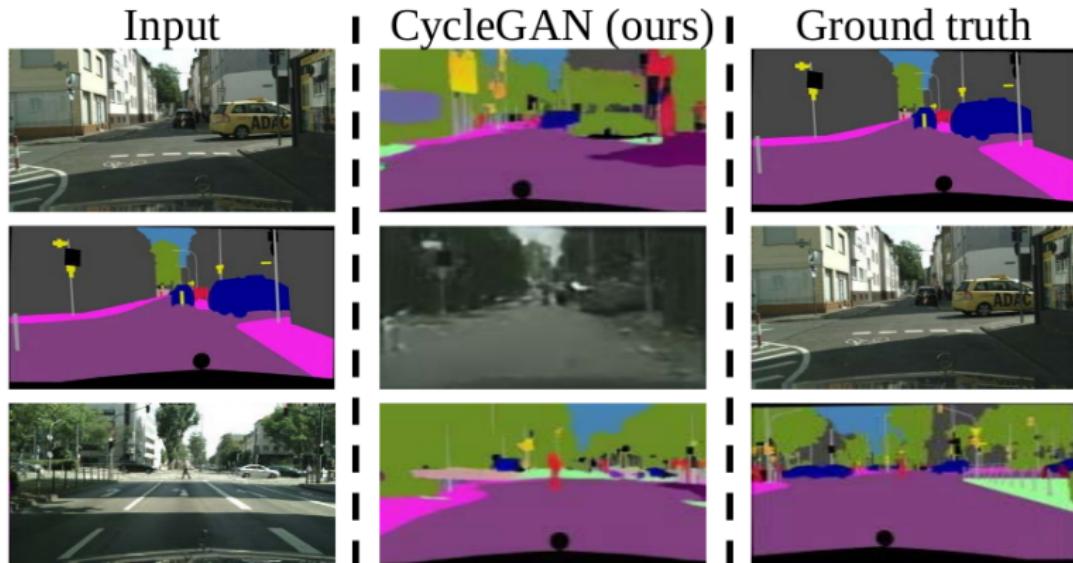
AMT perceptual studies

Amazon Mechanical Turk — Яндекс.Толока от Амазона;



Оценка с помощью CNN

- Хотим, чтобы картинка, получившаяся из цепочки label map → фото → label map, и исходный label map были похожи;
- Пусть CNN говорит, что происходит на исходном label map и на сгенерированном;
- Если описания совпадают, значит успех;



Semantic segmentation metrics

Для photo→labels считаем per-pixel accuracy, per-class accuracy, class IOU

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Additional results on paired datasets

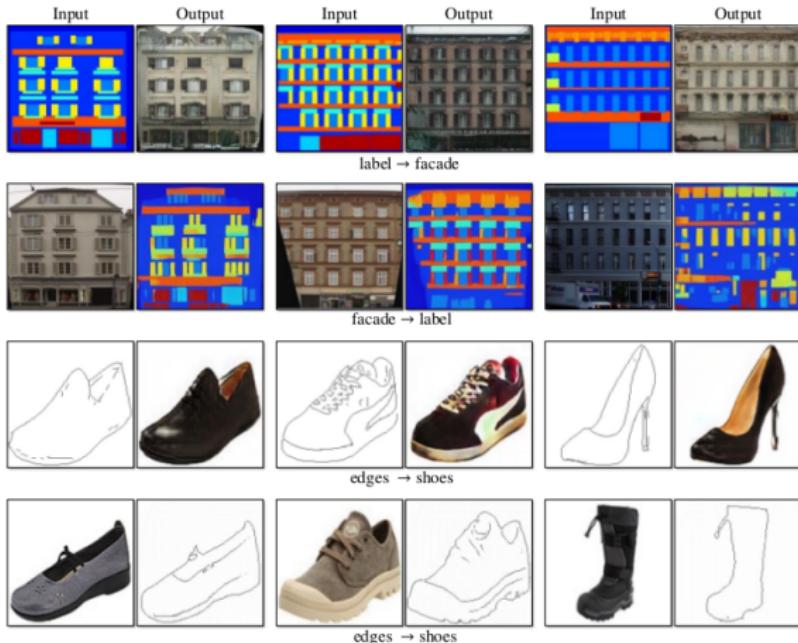


Figure 8: Example results of CycleGAN on paired datasets used in “pix2pix” [21] such as architectural labels \leftrightarrow photos and edges \leftrightarrow shoes.

Сравнение с другими методами

Loss	Map → Photo	Photo → Map
	% Turkers labeled <i>real</i>	% Turkers labeled <i>real</i>
CoGAN [30]	0.6% ± 0.5%	0.9% ± 0.5%
BiGAN/ALI [8, 6]	2.1% ± 1.0%	1.9% ± 0.9%
SimGAN [45]	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 [30]	0.40	0.10	0.06
BiGAN/ALI [8, 6]	0.19	0.06	0.02
SimGAN [45]	0.20	0.10	0.04
Feature loss + GAN	0.06	0.04	0.01
CycleGAN (ours)	0.52	0.17	0.11
pix2pix [21]	0.71	0.25	0.18

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

Сравнение с другими методами

Loss	Per-pixel acc.	Per-class acc.	Class IOU
CoGAN [30]	0.45	0.11	0.08
BiGAN/ALI [8, 6]	0.41	0.13	0.07
SimGAN [45]	0.47	0.11	0.07
Feature loss + GAN	0.50	0.10	0.06
CycleGAN (ours)	0.58	0.22	0.16
pix2pix [21]	0.85	0.40	0.32

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

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

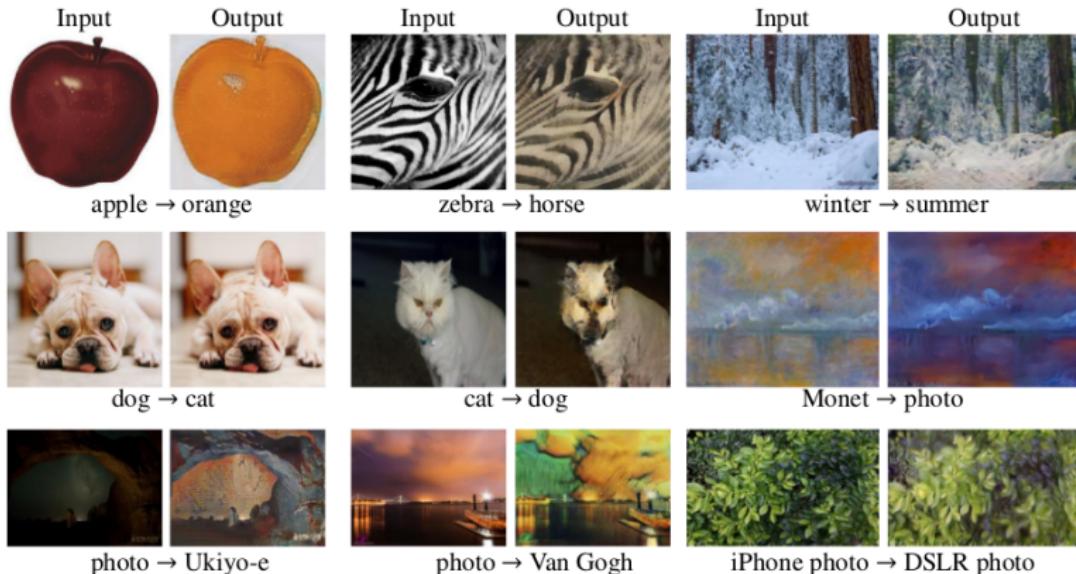
Table 4: Ablation study: FCN-scores for different variants of our method, evaluated on Cityscapes labels→photo.

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

Table 5: Ablation study: classification performance of photo→labels for different losses, evaluated on Cityscapes.

Основные сложности

- Когда нужны геометрические преобразования ($\text{dog} \rightarrow \text{cat}$);
- Иногда меняет ярлыки для зданий и деревьев в задании $\text{photos} \rightarrow \text{labels}$, решение – добавление мягкого supervised



Основные сложности



Horse → Zebra



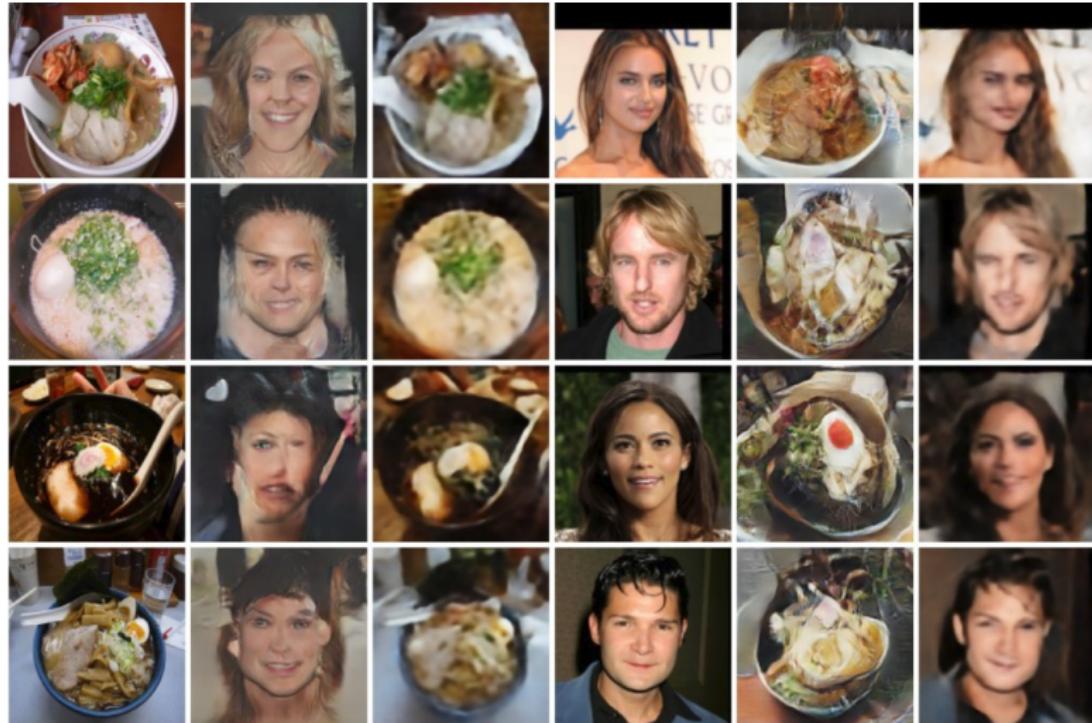
ImageNet “wild horse” training images

Когда обучался на датасете, содержащем только диких лошадей без наездников

Collection style transfer



Object transfiguration



Object transfiguration



Season transfer



winter Yosemite → summer Yosemite



summer Yosemite → winter Yosemite

Photo generation from paintings

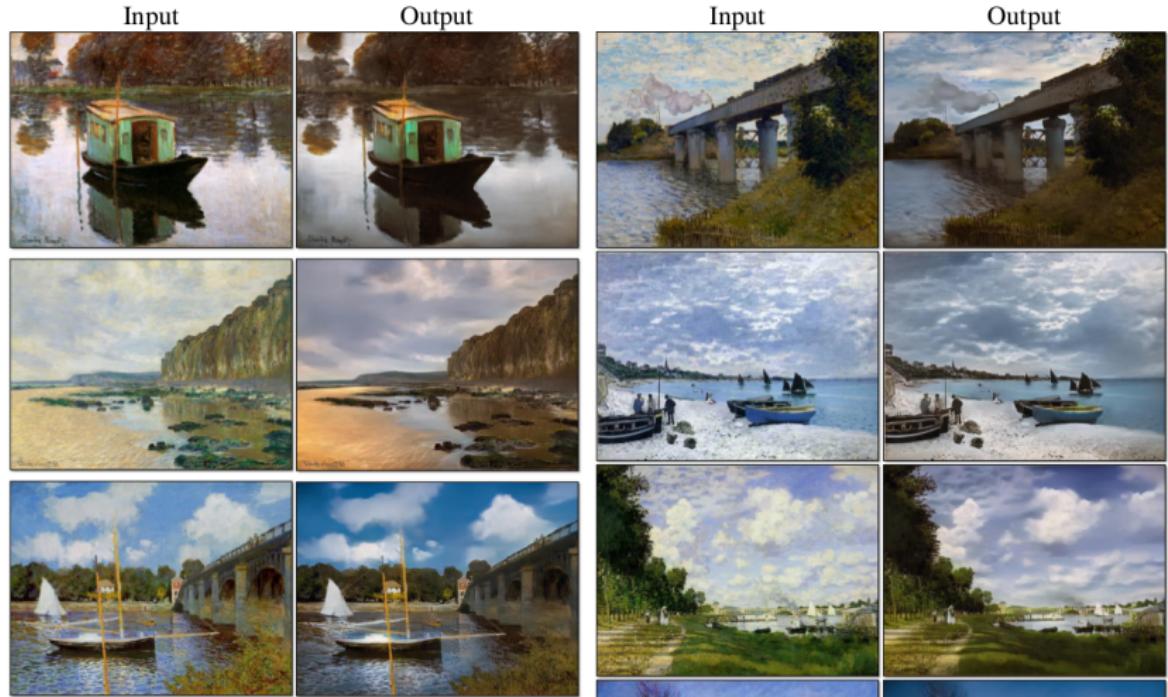


Photo enhancement



Figure 14: Photo enhancement: mapping from a set of iPhone snaps to professional DSLR photographs, the system often learns to produce shallow focus. Here we show some of the most successful results in our test set – average performance is considerably worse. Please see our website for more comprehensive and random examples.

Источники

- <https://arxiv.org/abs/1703.10593>
- <https://junyanz.github.io/CycleGAN/>
- <https://youtu.be/4sZsx4FpMxg> – human2vegetable