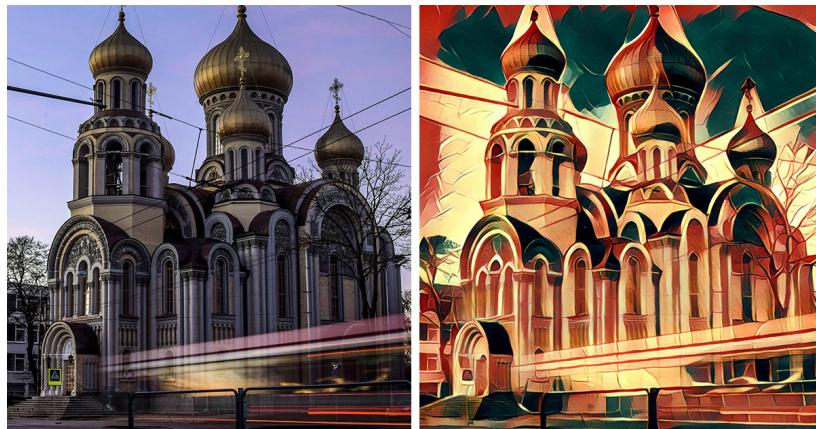


# Unpaired Image-to-Image Translation using Cycle- Consistent Adversarial Networks

# Image-to-Image Translation



$$x \in X \rightarrow y \in Y$$



$$x \in X \rightarrow y \in Y$$

Collection style transfer

# Image-to-Image Translation



Horse to zebra

Picture to semantic  
labels

**Проблема:** выборки paired-data либо вообще отсутствуют, либо их получение очень дорогое. Мы вынуждены работать с unpaired выборками.



# GAN: Generative Adversarial Network

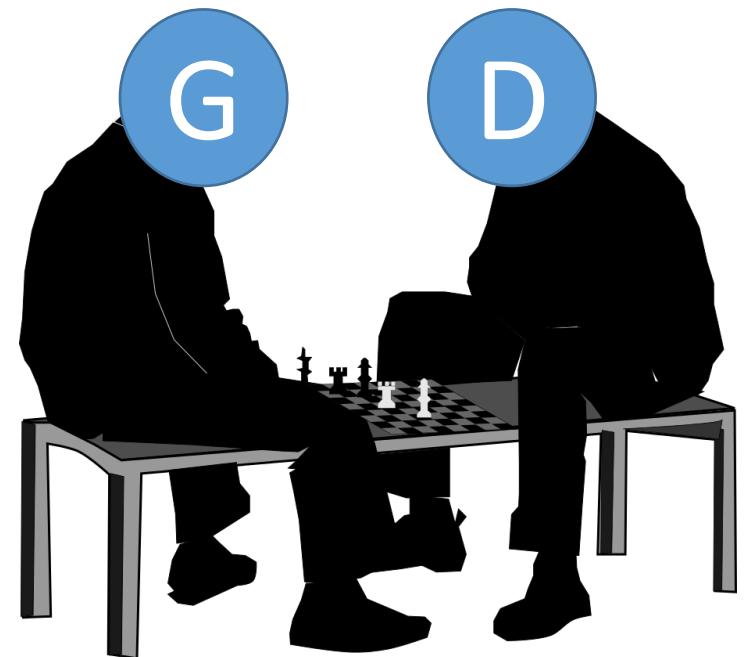
Задача: выучить отображение  $G: X \rightarrow Y$

Генератор vs. Дискриминатор

Генератор  $G: X \rightarrow Y$

Дискриминатор  $D_Y: Y \rightarrow [0, 1]$

Дана выборка настоящих картинок



# GAN: Generative Adversarial Network

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

# GAN: Generative Adversarial Network

С точки зрения **дискриминатора**

$$\underbrace{L(G, D, X, Y)}_{\max} = \mathbb{E}_{y \sim p_{data}(y)} [\log D(y)] + \mathbb{E}_{x \sim p_{data}(x)} [\log(1 - D(G(x)))]$$

# GAN: Generative Adversarial Network

С точки зрения генератора

$$\underbrace{L(G, D, X, Y)}_{\min} = \mathbb{E}_{y \sim p_{data}(y)} [\log D(y)] + \mathbb{E}_{x \sim p_{data}(x)} [\log(1 - D(G(x)))]$$

0

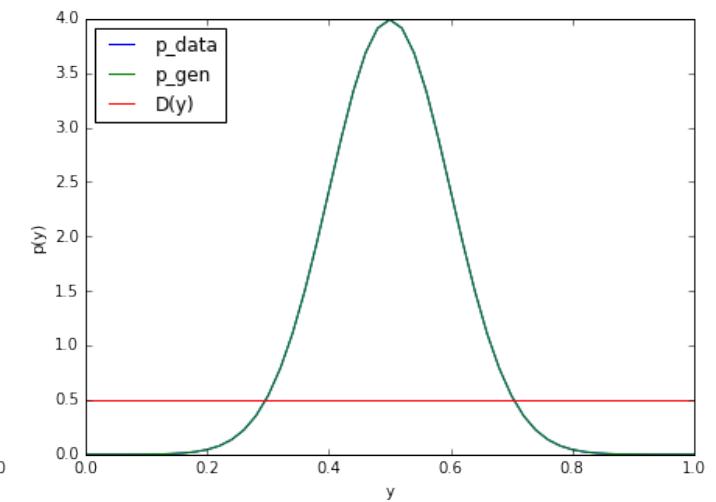
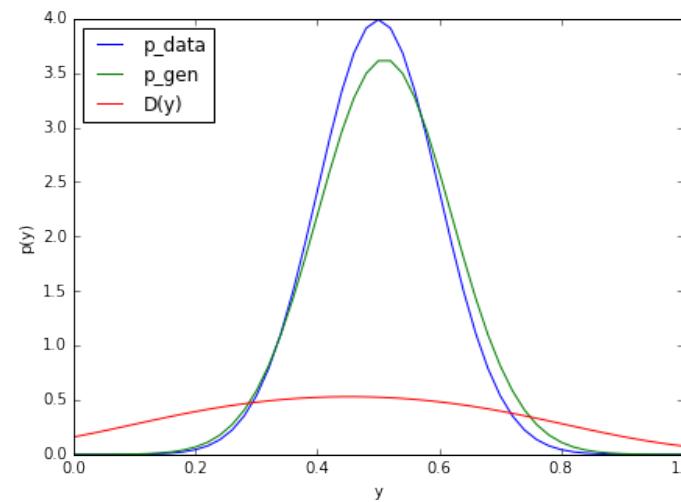
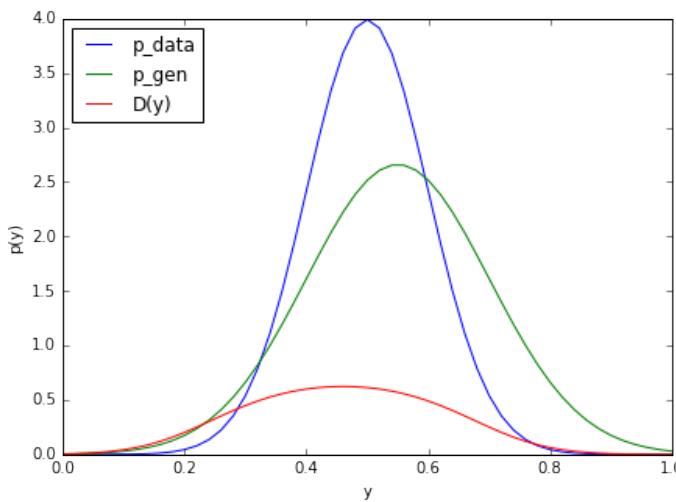
Оптимизационная задача

$$G^* = \arg \min_G \max_{D_Y} L(G, D_Y, X, Y)$$

# GAN: Generative Adversarial Network

Выведем правило поведения дискриминатора для выхода  $y$ . Возьмем производную loss'a по  $D(y)$  и приравняем к нулю. Получаем

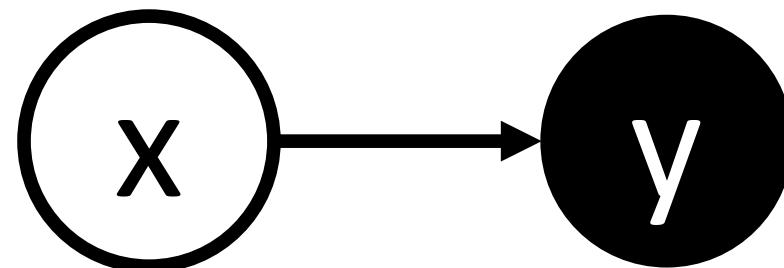
$$D^*(y) = \frac{p_{data}(y)}{p_{data}(y) + p_{model}(y)} \Rightarrow p_{model}^*(y) = p_{data}(y)$$



# GAN: Generative Adversarial Network

**Вывод:** стандартный GAN способен выучить генератор  $G$  такой, что производимые им выходы будут распределены одинаково с  $p(Y)$ .

**Проблема:** в нашем objective вход  $x$  и выход  $y$  никак не связываются. Поэтому, для множества входов из  $X$  сеть может выучить любую случайную перестановку выходов из  $Y$ .



## Попытки решения: pixel loss

- Связем вход и выход с помощью pixel loss:

$$L_{pixel}(G) = \mathbb{E}_{x \sim p_{data}(x)} \|x - G(x)\|_1$$

$$L_{total} = L_{GAN} + \gamma L_{pixel}$$

- Фактически мы заставляем сеть выучить identity-преобразование (при большом  $\gamma$ )
- Контрпример: инвертированные изображения



## Попытки решения: perceptual loss

- Пропустим вход и выход через глубинную нейросеть с уже готовыми весами (например, VGG-16). Будем минимизировать разницу между векторами признаков, взятых с предпоследнего слоя сети

$$L_{perceptual}(G) = \mathbb{E}_{x \sim p_{data}(x)} \|\text{deep}(x) - \text{deep}(G(x))\|_1$$

$$L_{total} = L_{GAN} + \gamma L_{perceptual}$$

- Результат сильно завязан на свойства используемой глубинной нейросети

# Cycle-Consistent Adversarial Networks

- Дополнительно к прямому преобразованию  $G$  выучим еще обратное преобразование  $F$ . В итоге мы обучаем 2 GAN'а

$$G: X \rightarrow Y, \quad F: Y \rightarrow X$$

- Будем требовать схожести входа  $x$  с результатом применения обратного преобразования к выходу  $y$ , и наоборот

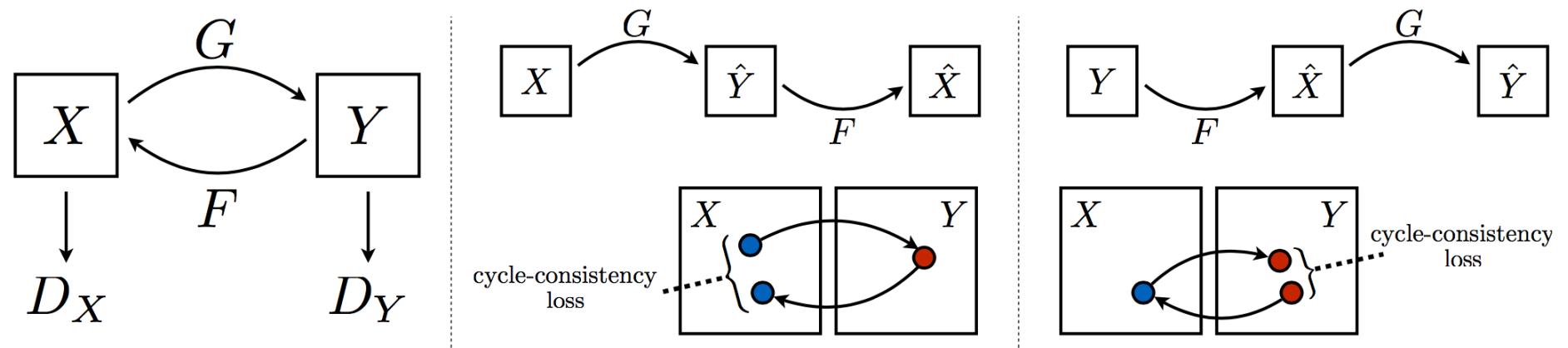
$$\begin{aligned} L_{cycle}(G, F) = & \mathbb{E}_{x \sim p_{data}(x)} \|x - F(G(x))\|_1 + \\ & + \mathbb{E}_{y \sim p_{data}(y)} \|y - G(F(y))\|_1 \end{aligned}$$

# Cycle-Consistent Adversarial Networks

- Итоговые loss и оптимизационная задача:

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

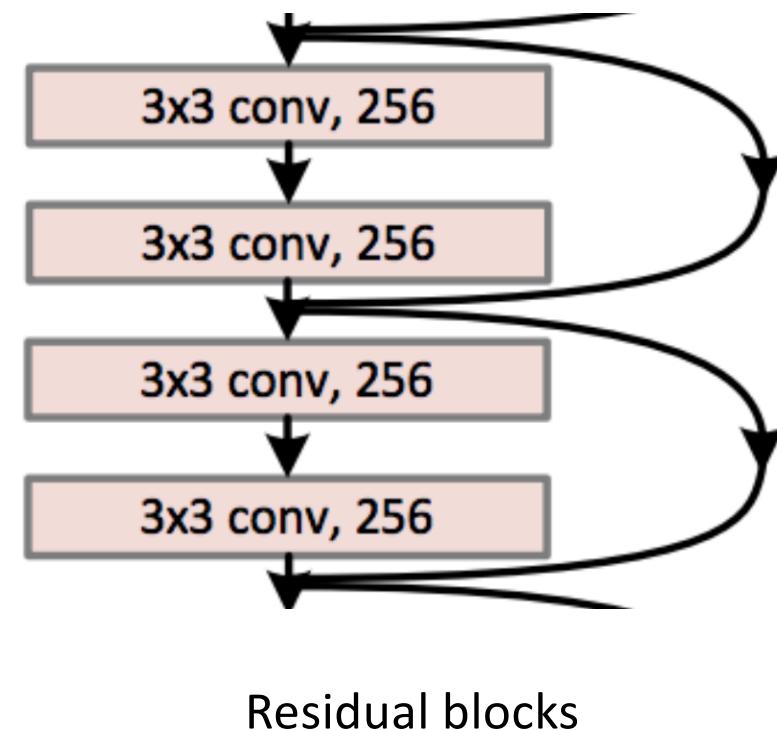
$$G^*, F^* = \arg \min_{F, G} \max_{D_X, D_Y} L(G, F, D_X, D_Y)$$



# Подробнее об архитектурах

## Генератор

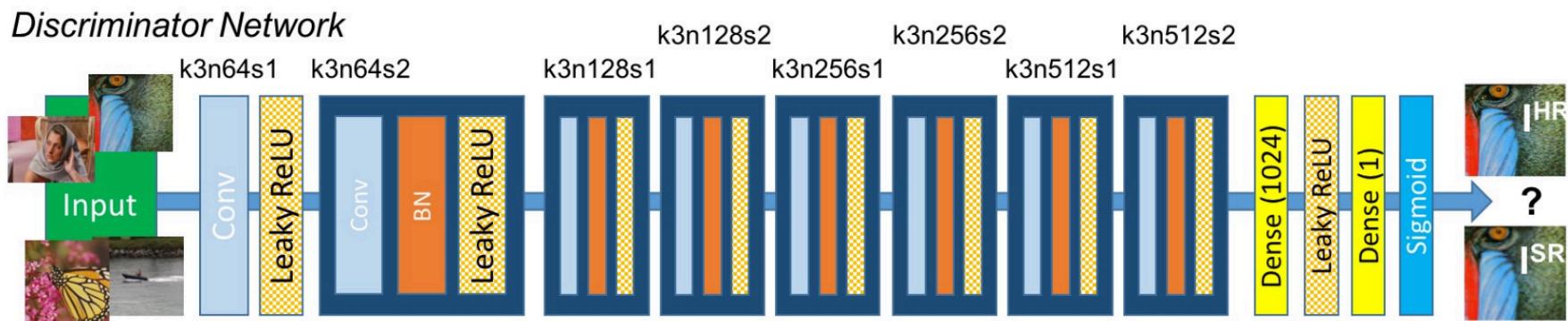
1. 2 stride-2 convolutions
2. 9 residual blocks: 2 convolutions in each of them
3. 2 transposed stride-1/2 convolutions  
+ Instance (contrast) normalization



# Подробнее об архитектурах

## Дискриминатор

Сверточная сеть, принимающая на вход патчи небольшого размера ( $70 \times 70$  патчей на одно изображение). Концентрируется на высокочастотных характеристиках изображения



# Обзор результатов

AMT perceptual studies: опросы респондентов на Amazon

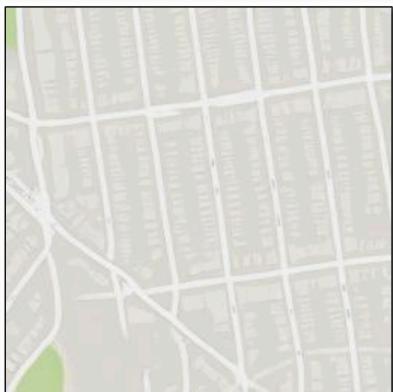
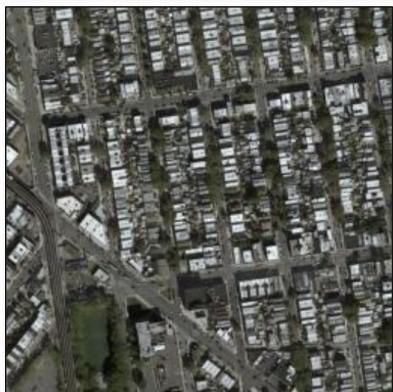


Фото местности со  
спутника в карту  
Google Maps и  
наоборот

Loss	Map → Photo % Turkers labeled <i>real</i>	Photo → Map % Turkers labeled <i>real</i>
CoGAN [27]	$0.6\% \pm 0.5\%$	$0.9\% \pm 0.5\%$
BiGAN [6, 5]	$2.1\% \pm 1.0\%$	$1.9\% \pm 0.9\%$
Pixel loss + GAN [41]	$0.7\% \pm 0.5\%$	$2.6\% \pm 1.1\%$
Feature loss + GAN	$1.2\% \pm 0.6\%$	$0.3\% \pm 0.2\%$
CycleGAN (ours)	<b><math>26.8\% \pm 2.8\%</math></b>	<b><math>23.2\% \pm 3.4\%</math></b>

# Обзор результатов

Модель тестировалась также на paired-data (классификация объектов на фото городских улиц), применялись стандартные для задачи метрики.



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)	<b>0.58</b>	<b>0.22</b>	<b>0.16</b>

# Обзор результатов



Фото



Моне



Ван Гог



Сезанн



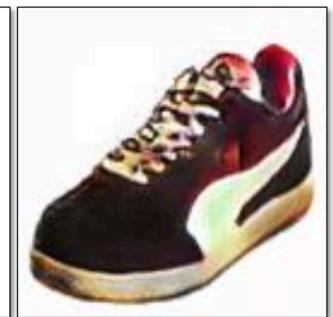
Укиё-э



Апельсины в яблоки



Фото с iPhone в фото  
с проф. камеры



Контур в обувь

## Список литературы

- **Основная статья** J.-Y. Zhu, T. Park, P. Isola, A. A. Efros. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks.* [[link](#)]
- I. Goodfellow. *NIPS 2016 Tutorial: Generative Adversarial Networks.* [[link](#)]
- J. Johnson, A. Alahi, L. Fei-Fei. *Perceptual Losses for Real-Time Style Transfer and Super-Resolution.* [[link](#)]
- C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. *Photo-realistic single image super-resolution using a generative adversarial network.* [[link](#)]