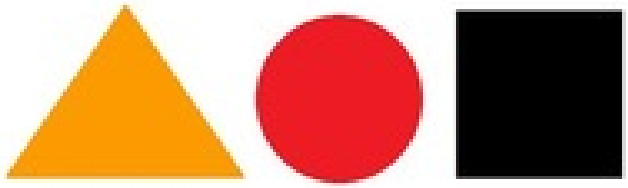


Школа Анализа Данных
Яндекса

Курс «Анализ изображений и видео, ч.2.»

Лекция №4
«Перенос стиля и синтез изображений»

Антон Конушин
Заведующий лабораторией компьютерной графики и мультимедиа
ВМК МГУ

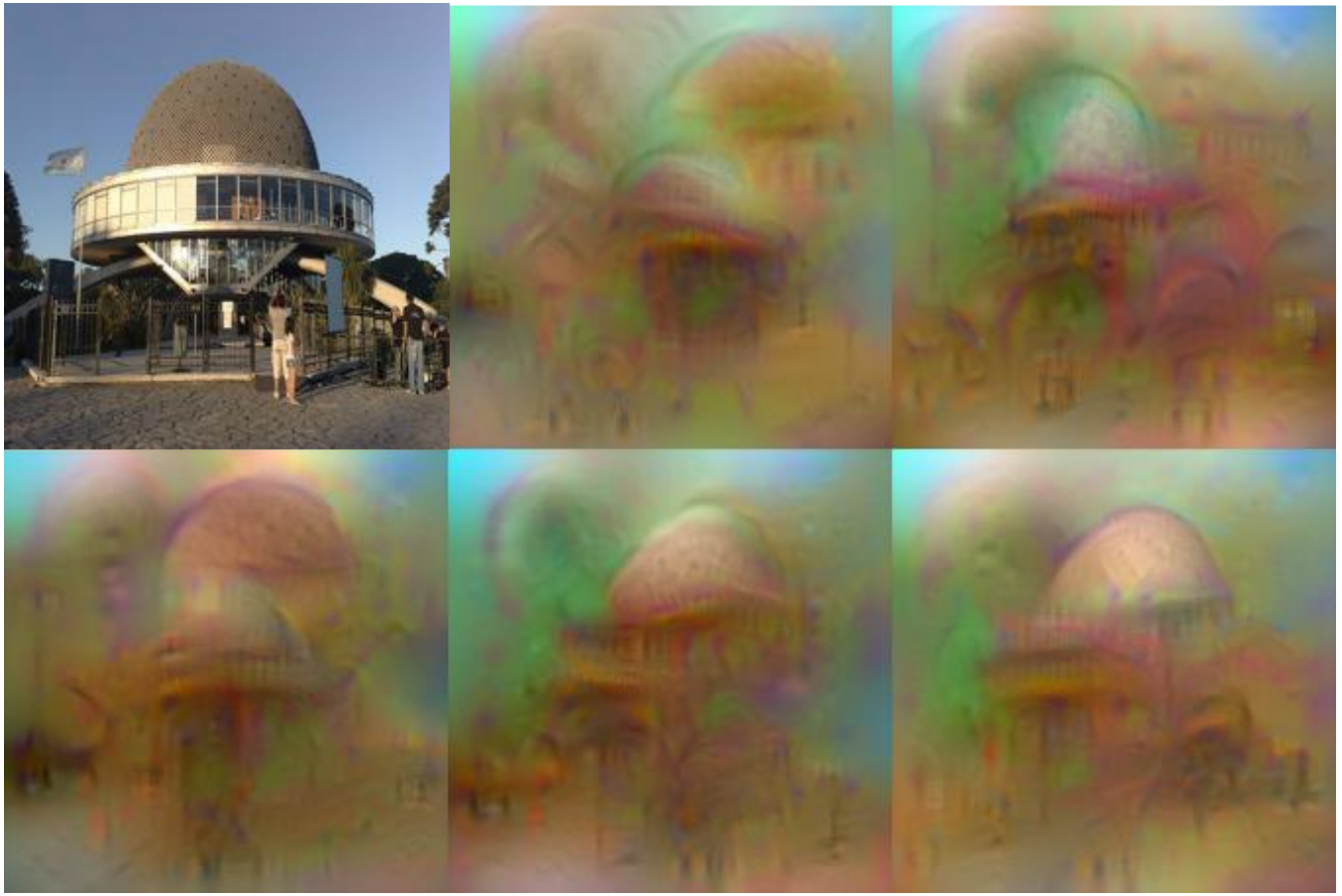
10 марта 2017 года

План

- Перенос стиля (style transfer)
- Генерация изображений с помощью GAN

Визуализация вектор-признака

Найдём такое изображение **x,** которое даёт такой же вектор-признак $\Phi_0 = \Phi(x_0)$ от изображения x_0



Mahendran and Vedaldi. Understanding Deep Image Representations by Inverting Them, 2014

Визуализация изображения по выходам

Процедура поиска:

- Инициализируем x белым шумом
- Будем оптимизировать следующий функционал:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^{H \times W \times C}}{\operatorname{argmin}} \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda \mathcal{R}(\mathbf{x})$$

$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

- R(x) – регуляризатор
- Оптимизируем градиентным спуском

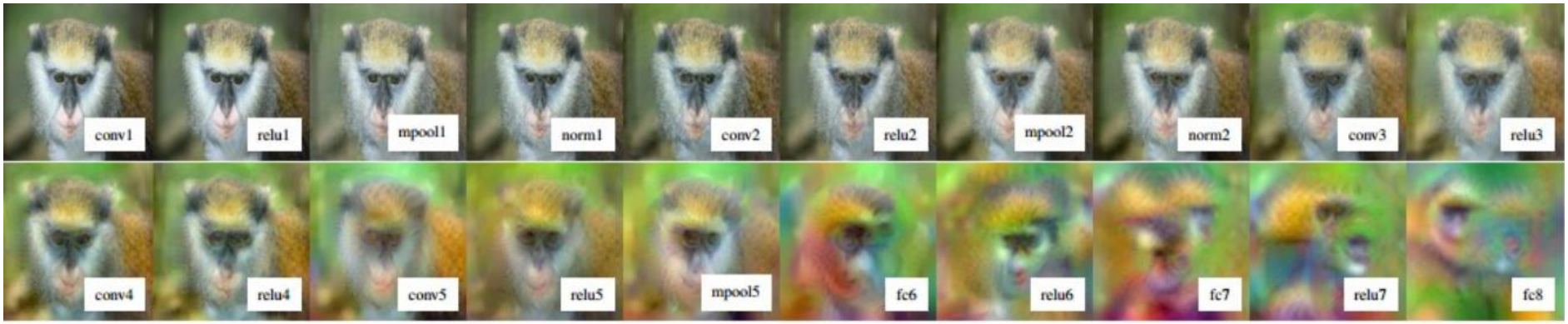
Реконструкция

С последнего свёрточного слоя

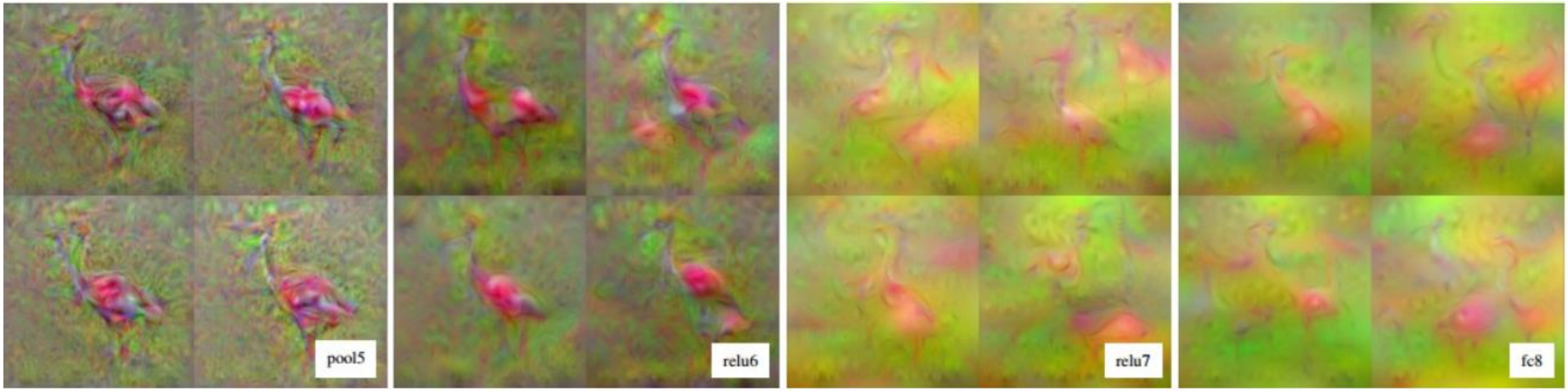
Пространственная информация во многом сохраняется

Реконструкция

С разных уровней



Множественные реконструкции (разные изображения с одинаковыми признаками)

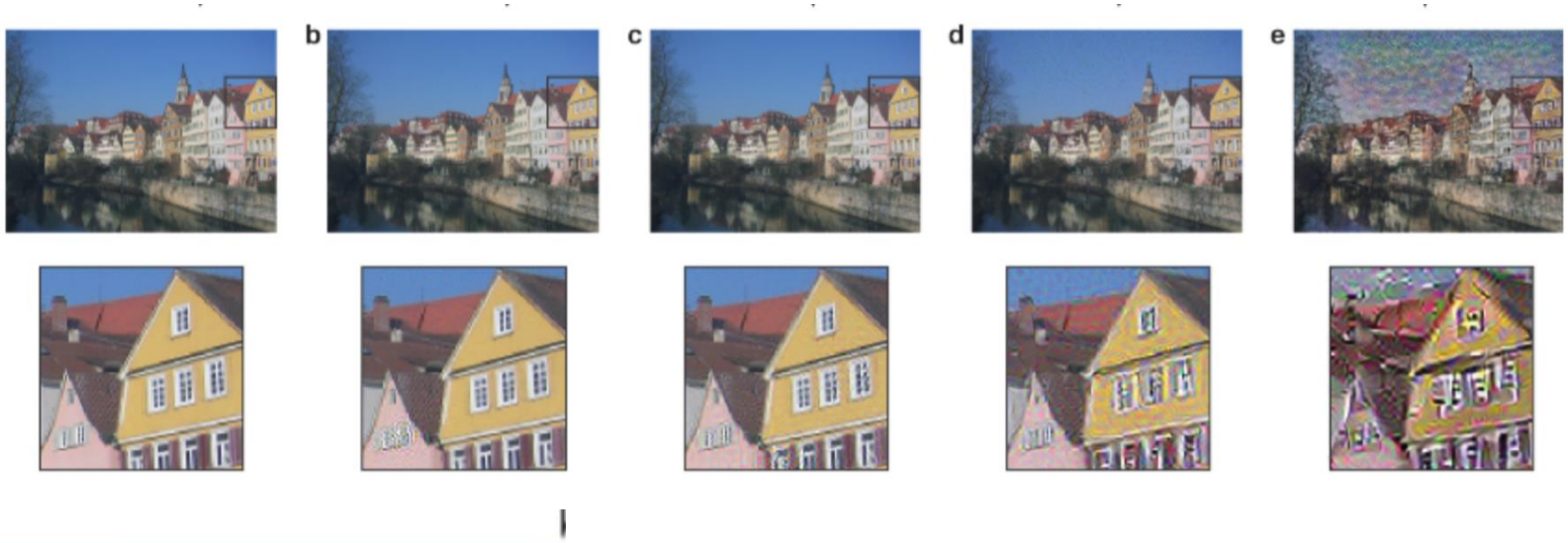


Перенос стиля

+

=

Реконструкция изображения



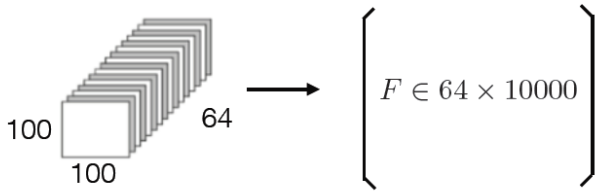
- Мы видели, что можно реконструировать исходное изображение по выходу любого слоя
- Пример – слои ‘conv1_1’, ‘conv2_1’, ‘conv3_1’, ‘conv4_1’, ‘conv5_1’ из нейросети VGG16
- Что можно сказать про содержимое слоёв на каждом из уровней?

Реконструкция стиля

- За описание «стиля» можно взять корреляцию откликов разных фильтров по всему изображению
- Можно вычислить по признакам первых слоёв «стиль» и попробовать реконструировать изображение с тем же стилем

Реконструкция стиля

- Стиль можно описать корреляцией откликов фильтров, записав матрицу Грама $G^l \in \mathcal{R}^{N_l \times N_l}$
- Где G^l_{ij} вычисляется как скалярное произведение откликов i-го и j-го фильтров:



$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}$$

- Генерируем изображения стиля, минимизируя среднеквадратичную разницу между матрицами Грама исходного изображения G и сгенерированного изображения A (или суммами по L первых слоёв)

$$E_l = \frac{1}{Norm} \sum_{i,j} (G^l_{ij} - A^l_{i,j})^2$$

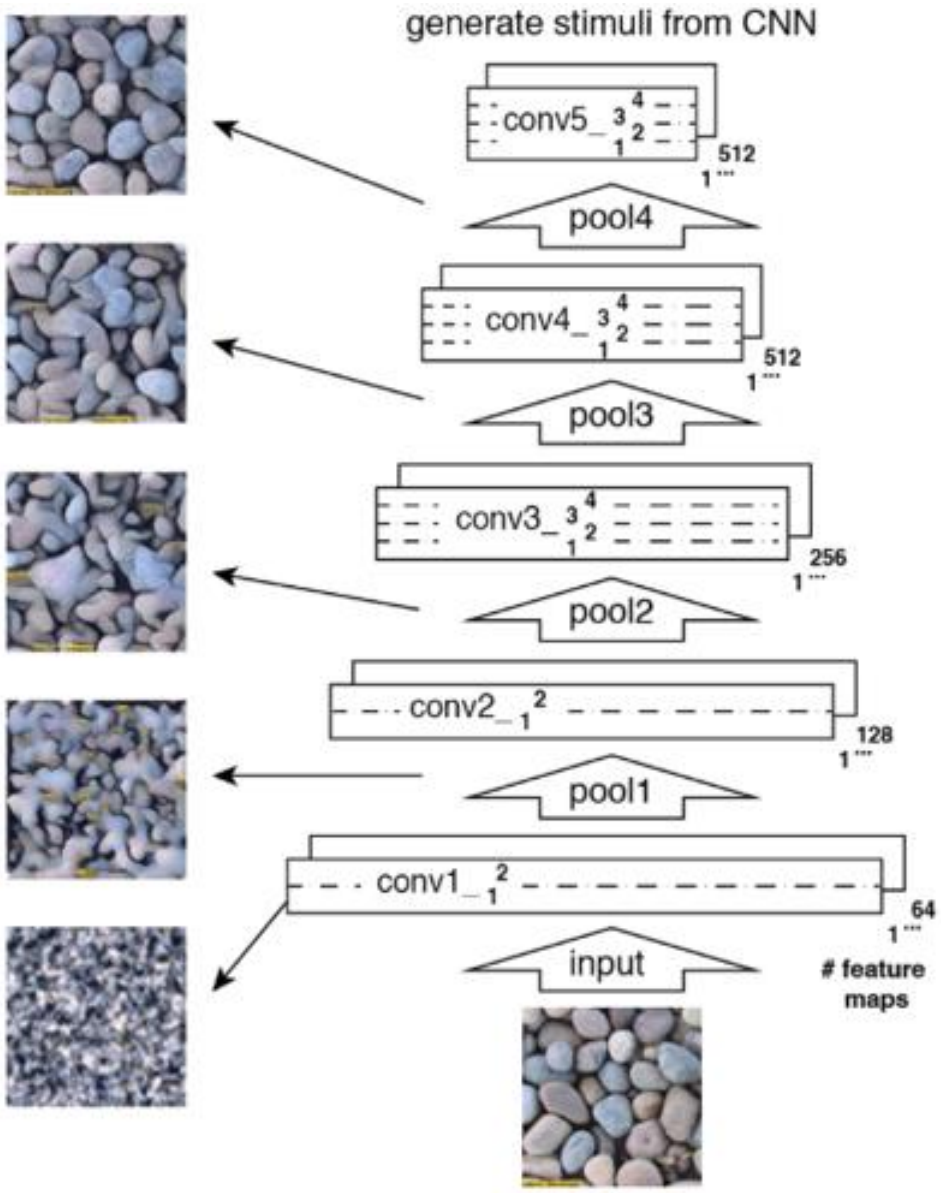
Cost for style reconstruction

$$Loss_{style} = \sum_{l=0}^{L} w_l E_l$$

Accumulate cost for lower layers

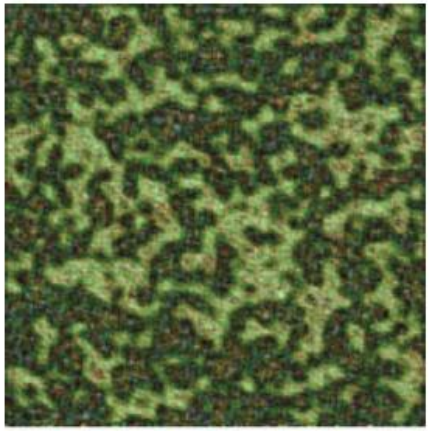
A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. Portilla, J., & Simoncelli, E. P. *Int. J. Comput. Vis.* 40, 49-70 (2000)
Texture Synthesis and The Controlled Generation of Natural Stimuli Using Convolutional Neural Networks. *Gatys, L. A., Ecker, A. S., & Bethge, M. NIPS (2015)*

Реконструкция текстур

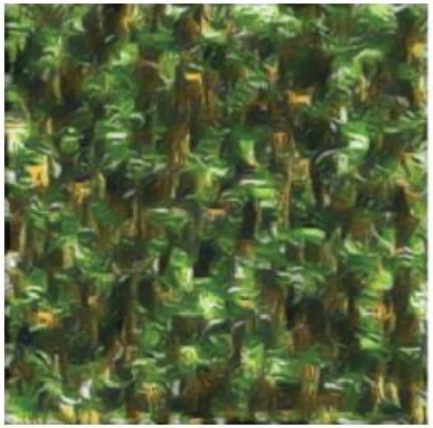


Пример реконструкции стиля

Original



Up to *Conv1_1* layer



Up to *Pool1* layer

Up to *Pool2* layer

Up to *Pool3* layer

Up to *Pool4* layer

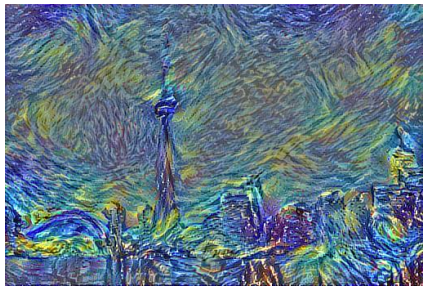
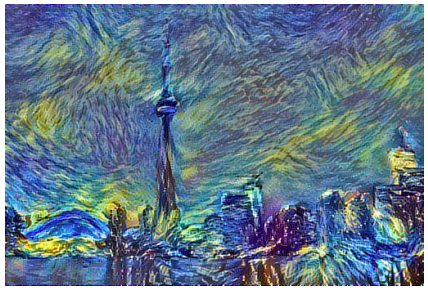
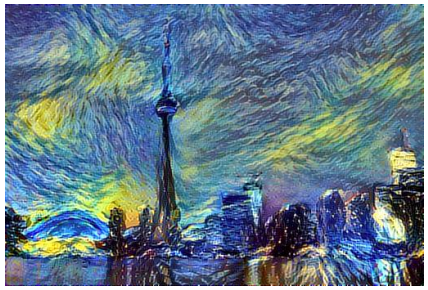
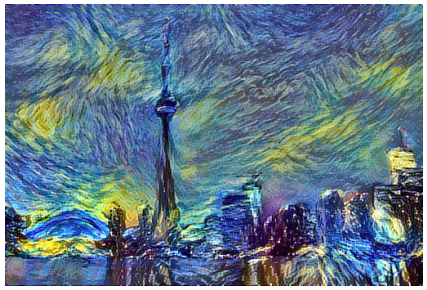
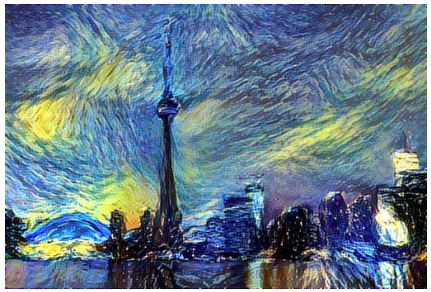
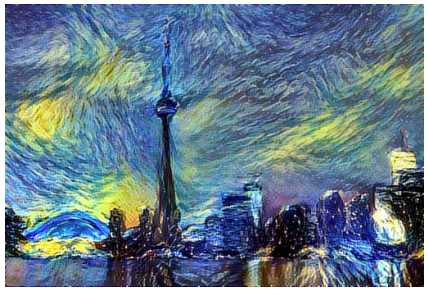
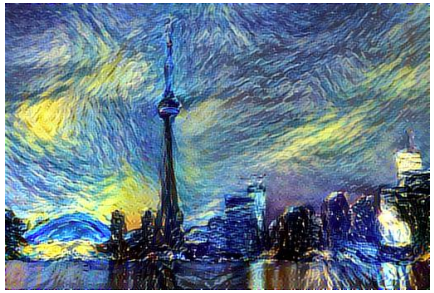
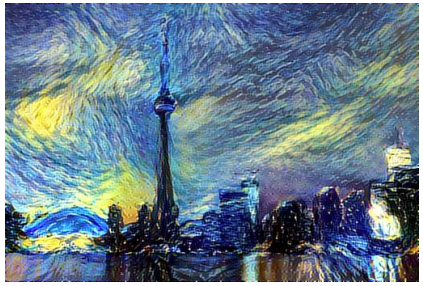
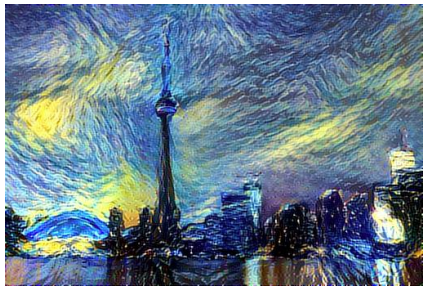
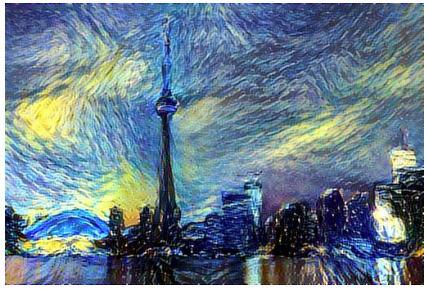
Ключевое наблюдение

- Содержание изображение и стиль оказываются разделимы
- Верхние слои больше описывают содержание изображения
- Нижние слои – стиль изображения
- Можем сгенерировать изображения, начав с белого шума, минимизировав градиентным спуском функционал:

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

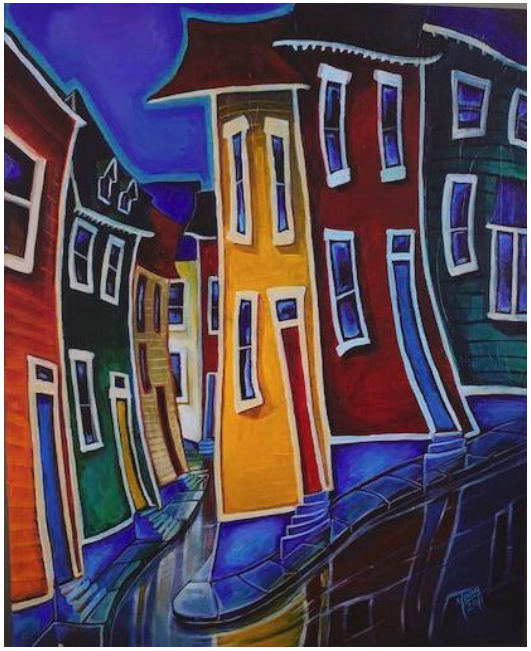
Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).

Визуализация работы

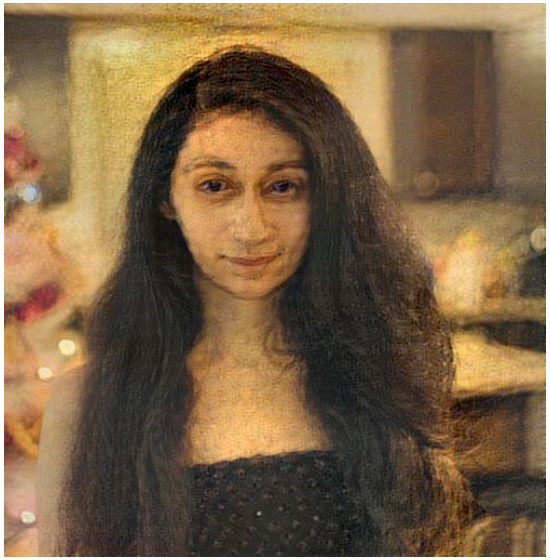











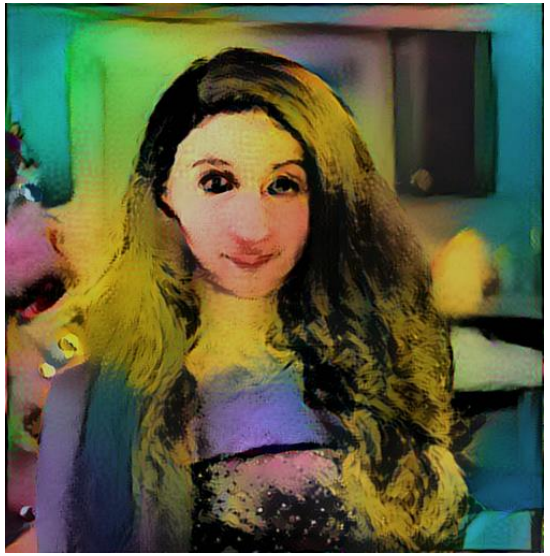
Результаты через
100 итераций
градиентного спуска

Пример работы

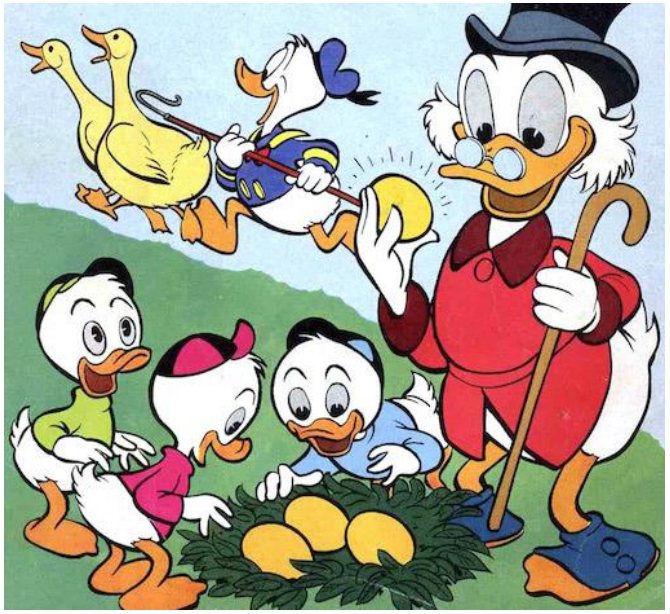


Пример работы



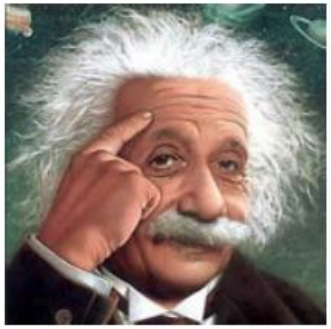


Пример работы



Соотношение стиля и содержания

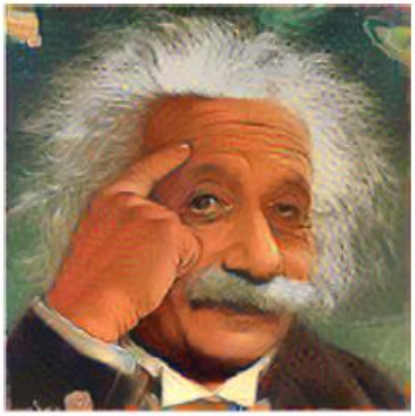
$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$



Used for *Content*

Used for *Style*

$\alpha/\beta = 10^{-3}$



$\alpha/\beta = 10^{-4}$

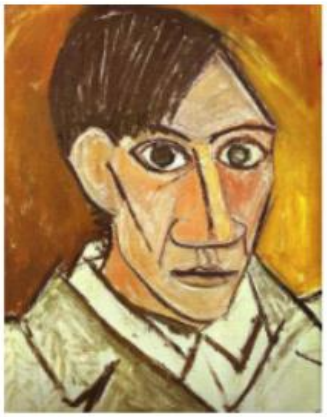
$\alpha/\beta = 10^{-5}$

Соотношение стиля и содержания

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

Decrease α/β

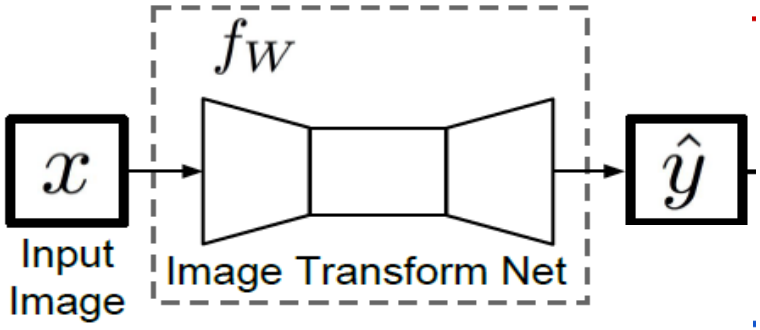
Used for *Content*



Used for *Style*

Генерация с помощью нейросети

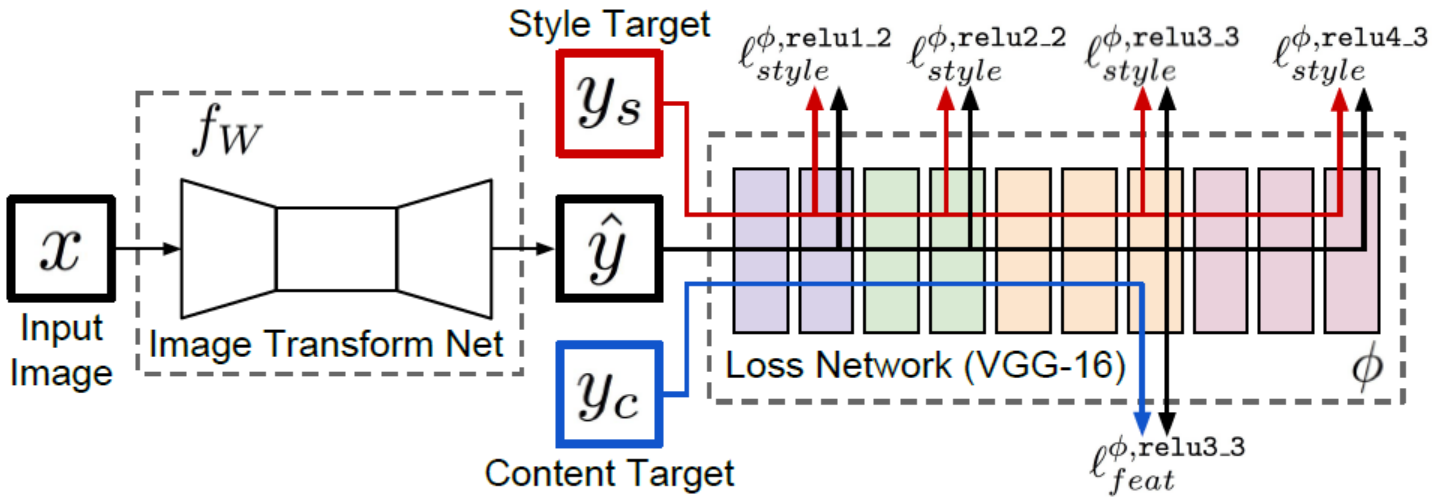
- Оптимизация с белого шума идёт медленно
- Идея: сделать нейросеть, преобразующую изображений в нужный стиль
- На входе и на выходе изображение



- Для каждого стиля нужно обучать свою собственную нейросеть
- Для subsampling не используем pooling, а свёртки с шагом

Johnson et. al. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. ECCV 2016

Обучение такой сети



- Воспользуемся предобученной на задаче классификации на imagenet сетью VGG-16
- Будем использовать её для извлечения признаков, и она будет зафиксирована при обучении трансформационной сети
- Через неё прогоняем изображения y_s (стиля), и y_c=x (содержание)

Composition VII, Wassily Kandinsky, 1913

Original Gatys et al. Ours

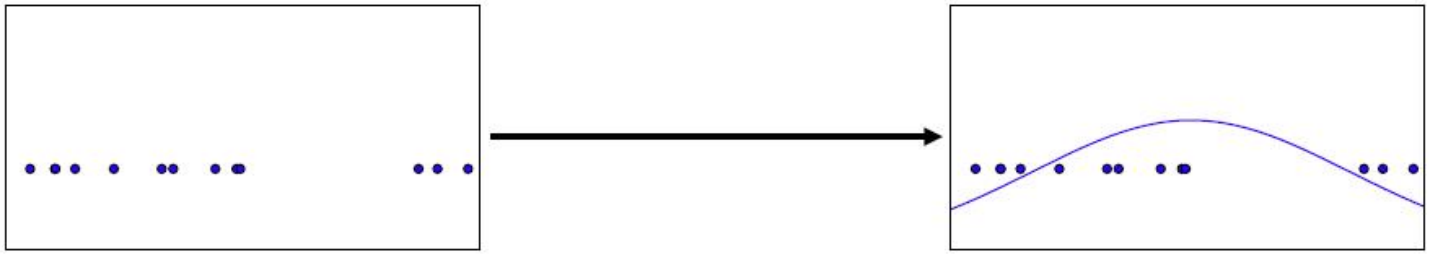
The Muse,
Pablo Picasso,
1935

Original Gatys et al. Ours

Generative Adversarial Networks

Порождающие модели

- Density estimation



- Sample generation

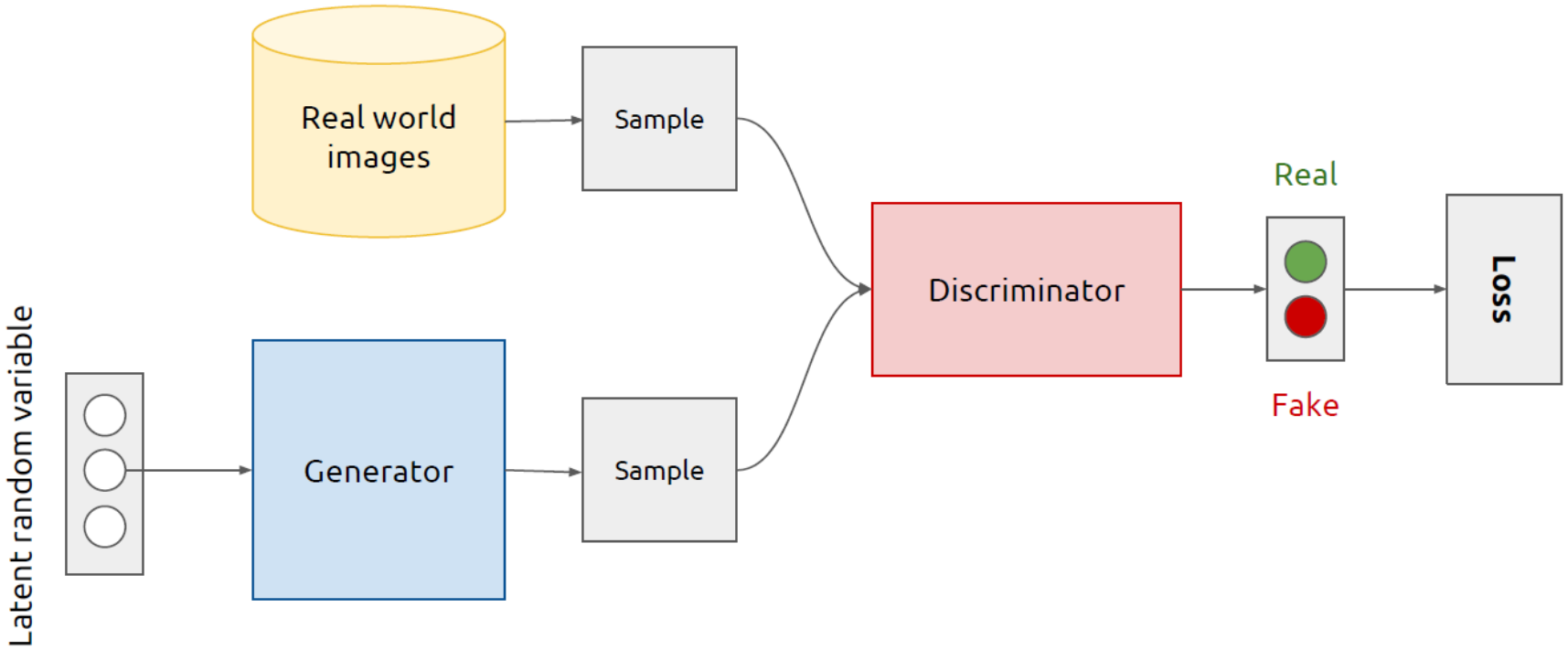
Training examples

Model samples

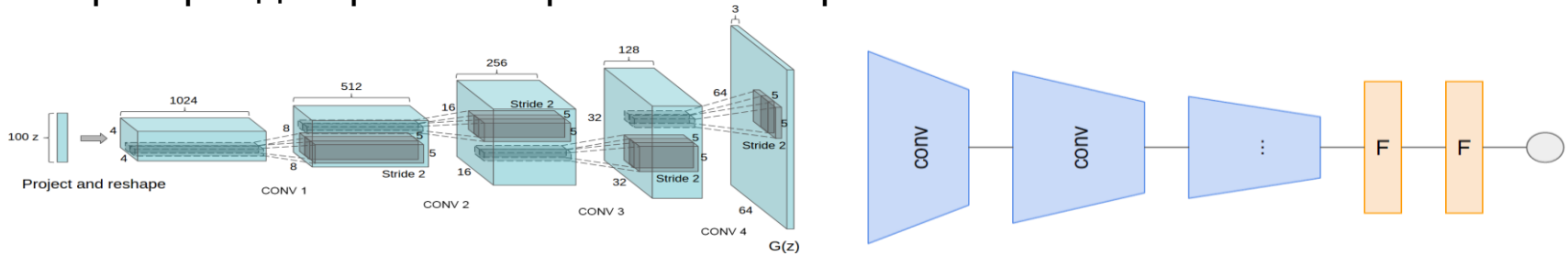
(Goodfellow 2016

Общая схема GAN

Сталкиваем генератор изображений из шума и дискриминатор:

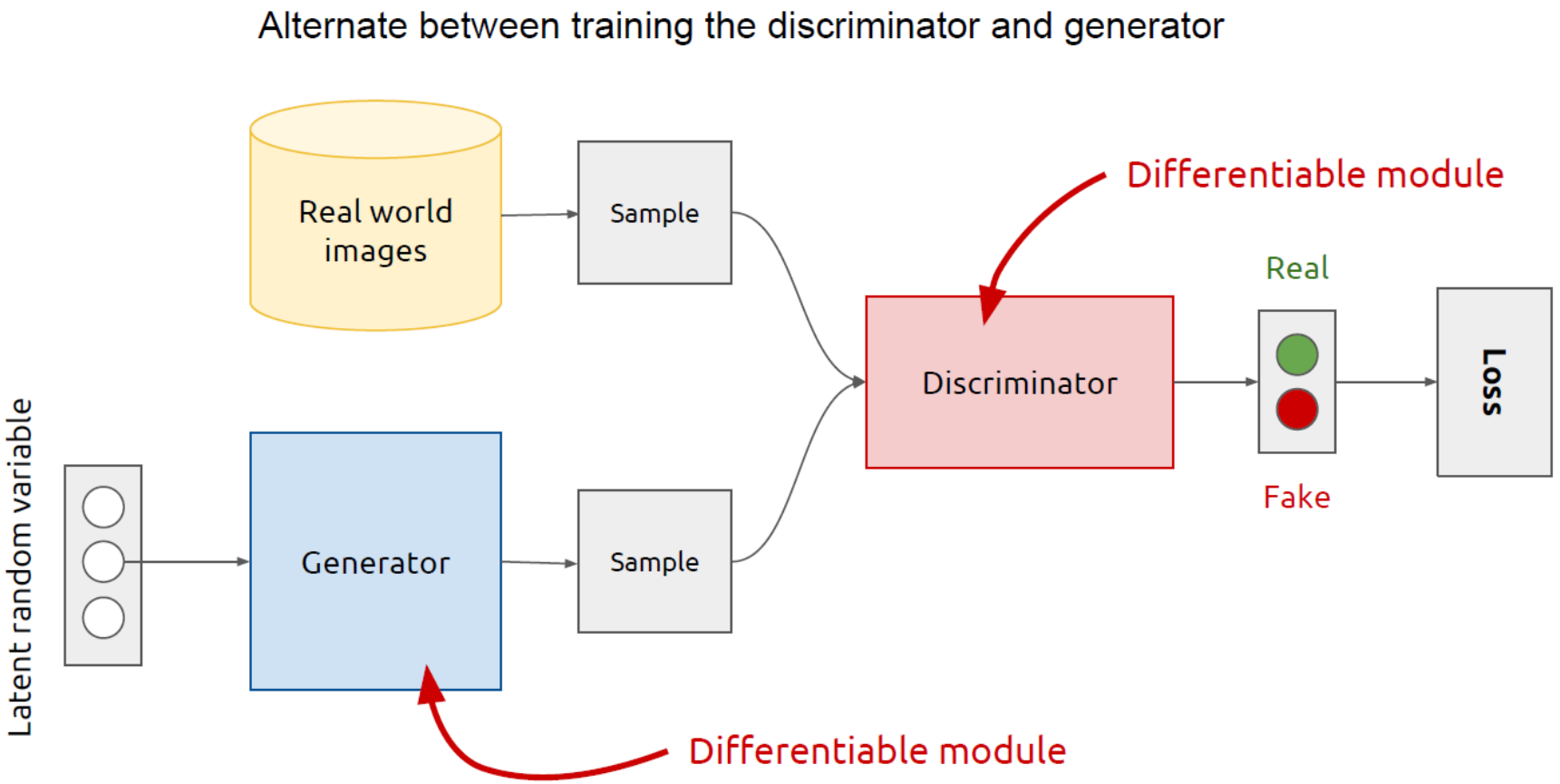


Генератор и дискриминатор обычно нейросети:

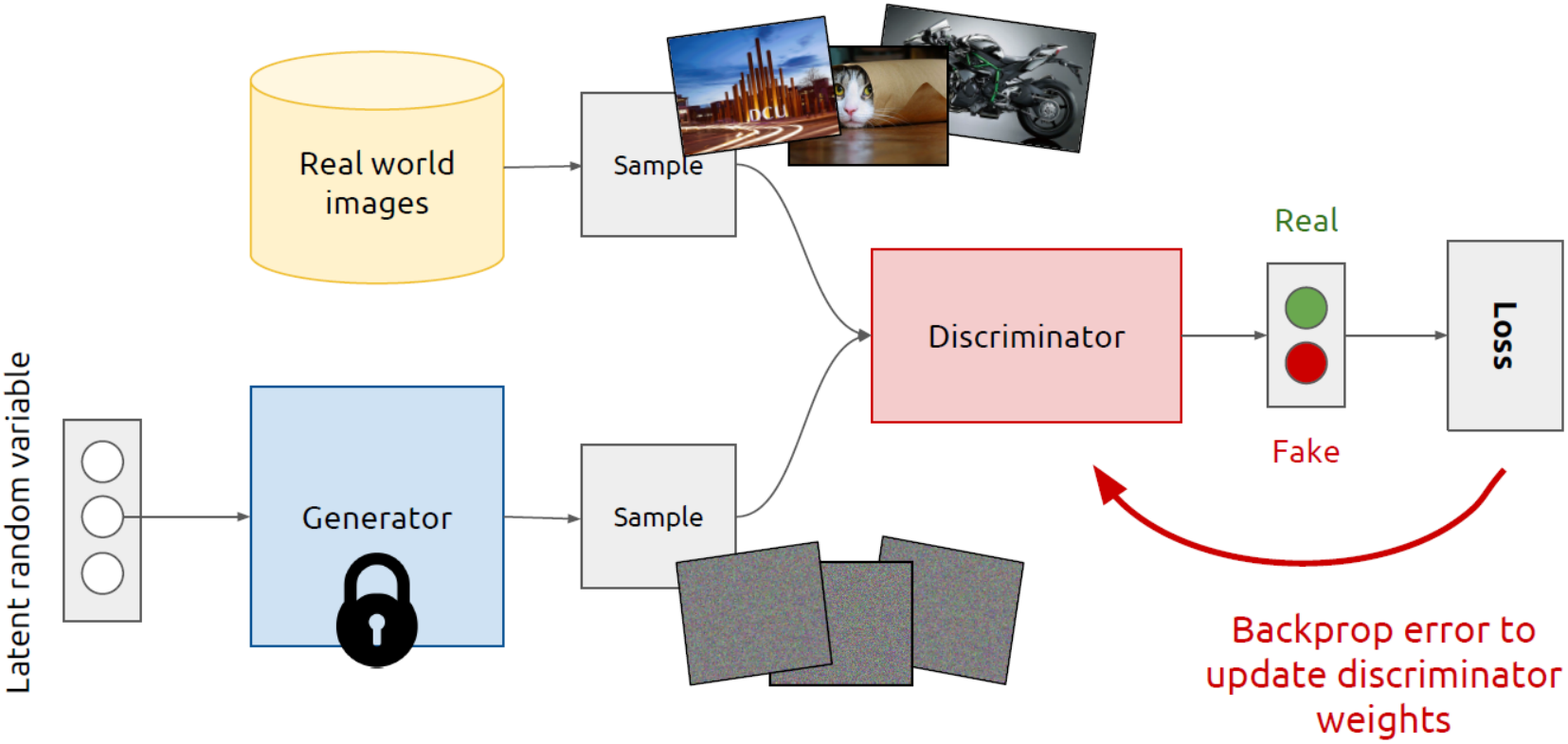


Ian Goodfellow et al, “Generative Adversarial Networks”, 2014

Обучение GAN

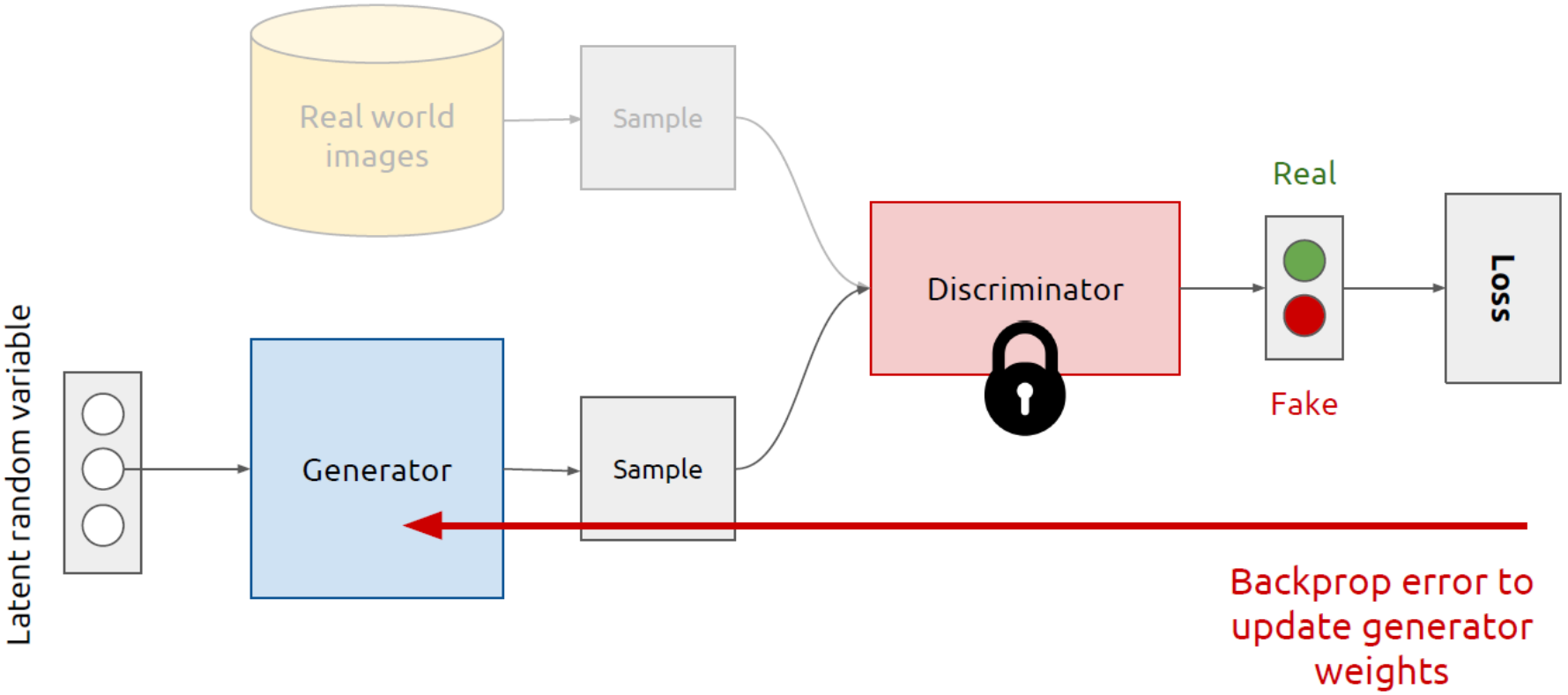


Обучение GAN



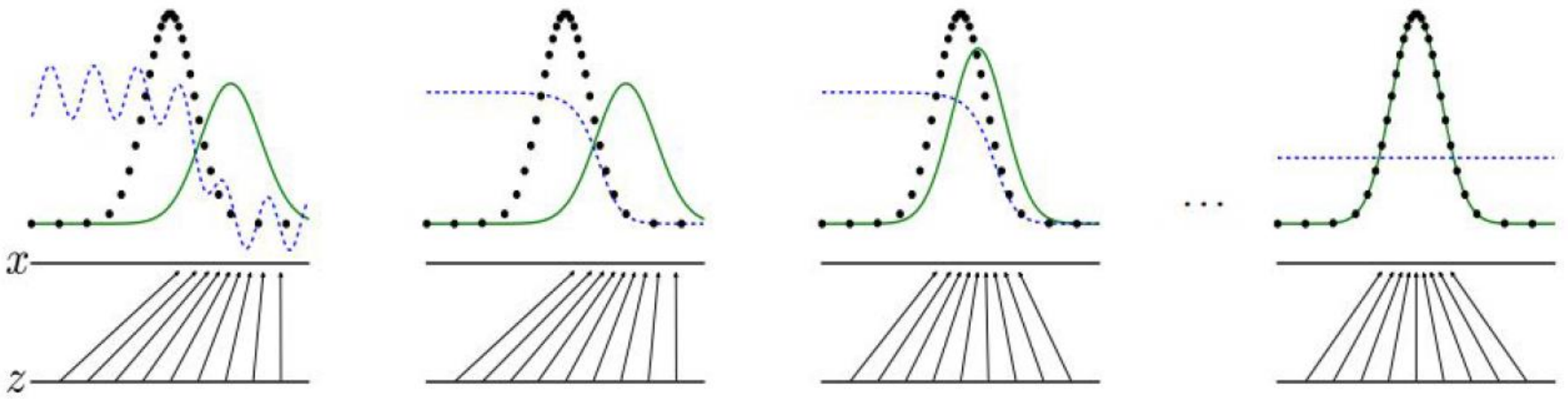
- Фиксируем веса генератора
- Берём примеры реальных изображений и синтезированных изображений
- Учим дискриминатор из различать

Обучение GAN



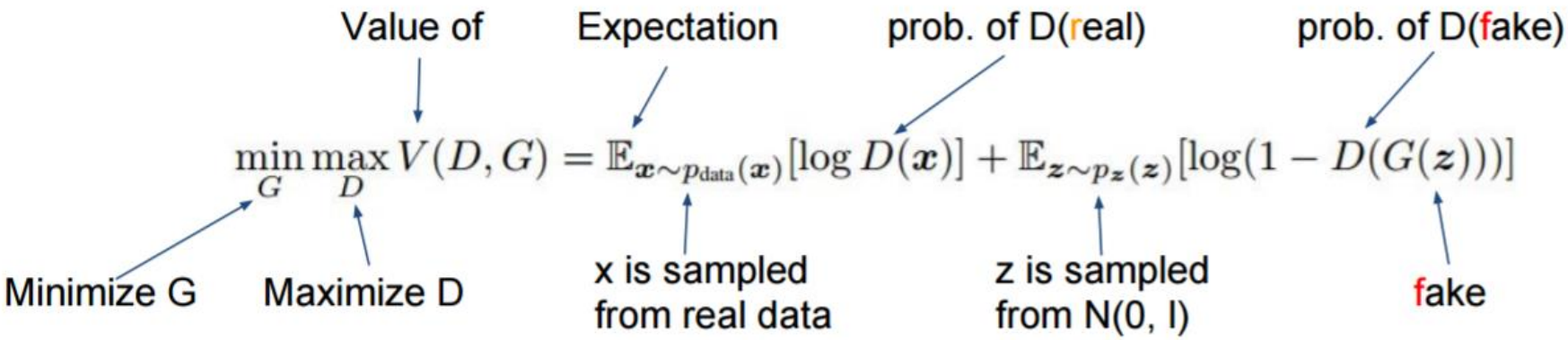
- Фиксируем веса дискриминатора
- Генерируем примеры генератором
- Распространяем ошибку от дискриминатора

Обучение GAN



- Чередование обучение дискриминатора повторяется до сходимости, которая может и не состояться
- В конце мы надеемся, что генератор научиться делать такие изображения, что дискриминатор не сможет их различить

Математическая формулировка цели



Обучение GAN

- Mathematical notation - discriminator

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

Maximize prob. of D(real)

Minimize prob. of D(fake)

BCE(binary cross entropy) with label 1 for real, 0 for fake.

(Practically, **CE** will be OK. or more plausible.)

Обучение GAN

- Mathematical notation - generator

$$\min_G V(D,G) = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \simeq \max_G \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(D(G(\boldsymbol{z})))]$$

Maximize prob. of D(fake)

BCE(binary cross entropy) with label 1 for fake.

(Practically, **CE** will be OK. or more plausible.)

Обучение GAN

- Mathematical notation - equilibrium

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \longrightarrow 0.5$$

$$\begin{aligned} C(G) &= \max_D V(G, D) \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}[\log(1 - D_G^*(G(\boldsymbol{z})))] \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] \\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(\boldsymbol{x})}{P_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x} \sim p_g}\left[\log \frac{p_g(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_g(\boldsymbol{x})}\right] \end{aligned}$$

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \middle\| \frac{p_{\text{data}} + p_g}{2}\right) + KL\left(p_g \middle\| \frac{p_{\text{data}} + p_g}{2}\right)$$

$$2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$$

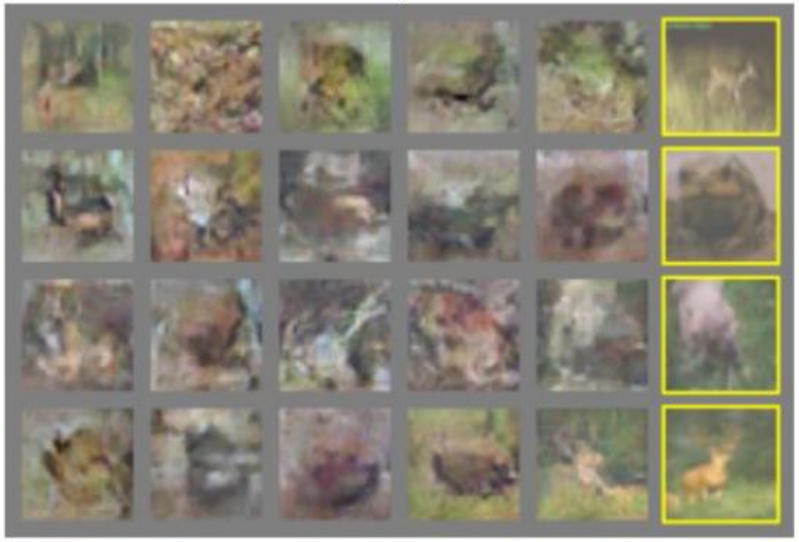
Jansen-Shannon divergence

Результаты

a)

b)

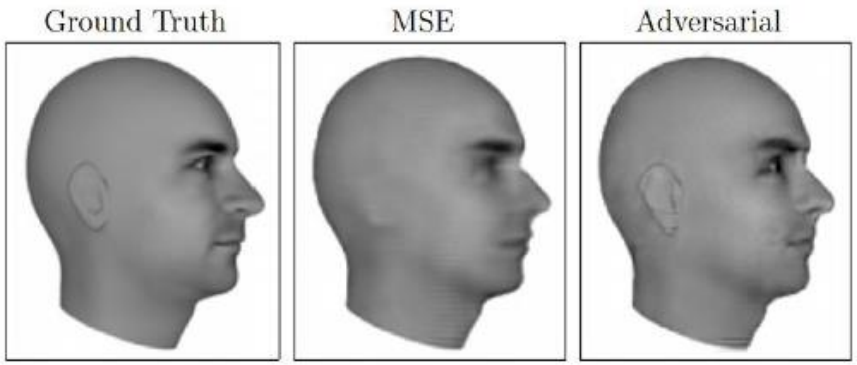
c)



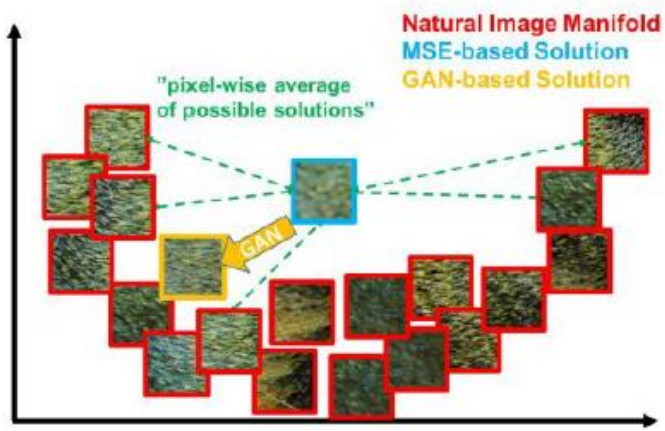
d)

Четкость результата

- Why blurry and why sharper ?



From Ian Goodfellow, Deep Learning (MIT press, 2016)

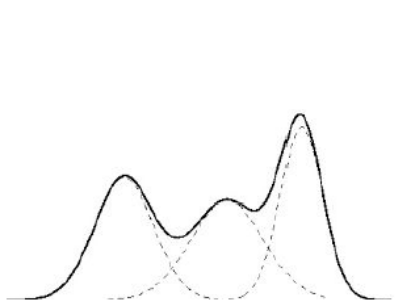


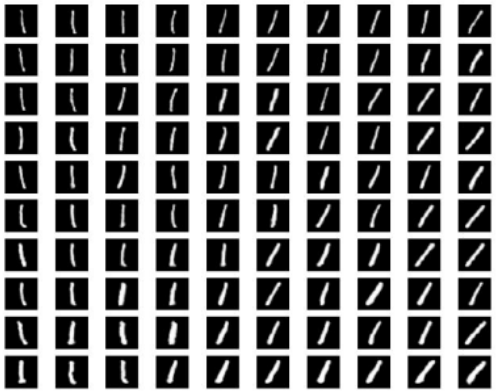
From Christian et al, Photo-realistic single image super-resolution using generative adversarial networks, 2016

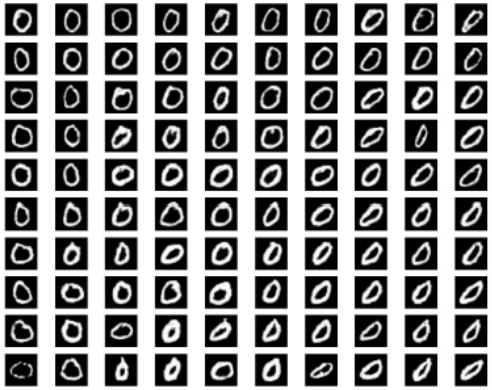
Обучение GAN

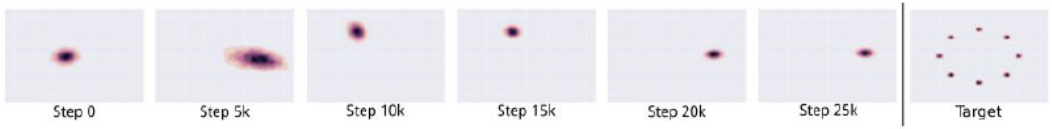
- No guarantee to equilibrium
  - **Mode collapsing**
  - Oscillation
  - No indicator when to finish
- All generative model
  - Evaluation metrics (Human turing test)
  - Overfitting test (Nearest Neighbor) → GAN is robust !!!
  - Diversity test

Основные проблемы GAN







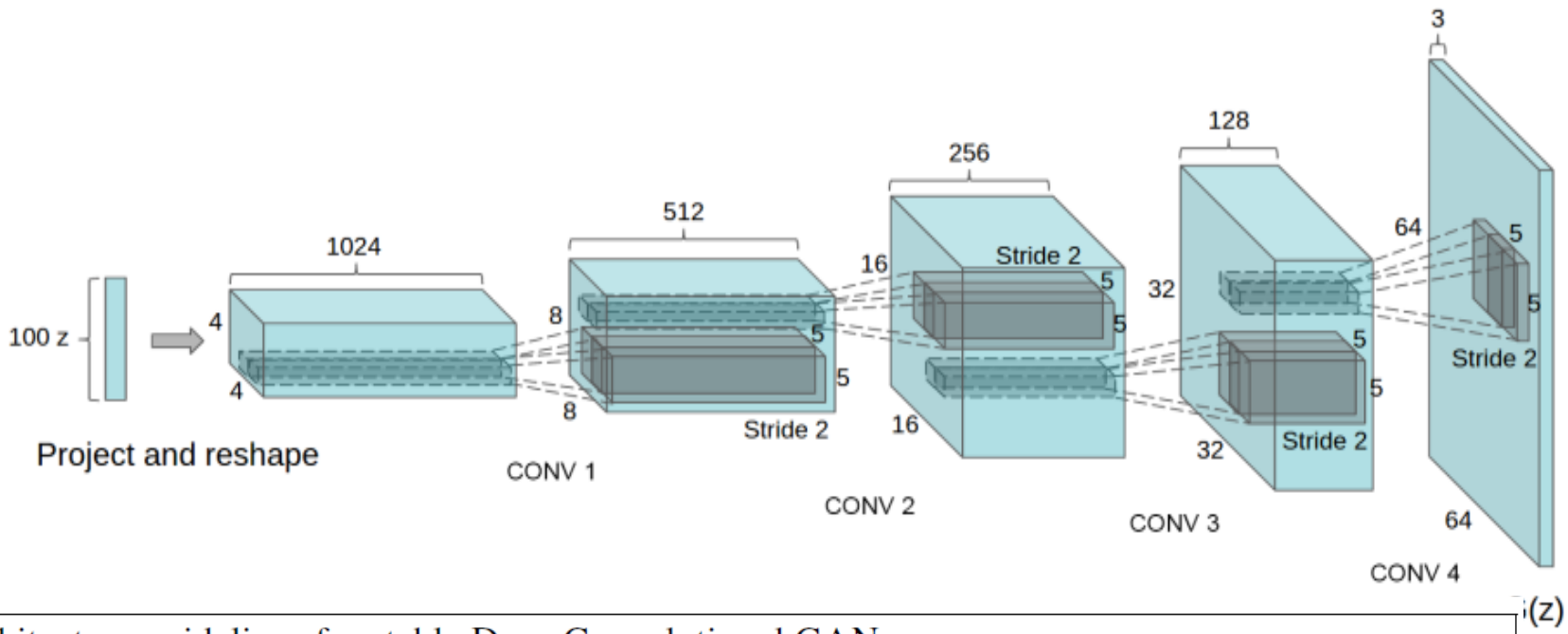


mode collapsing

oscillating

- https://github.com/soumith/ganhacks

DCGAN



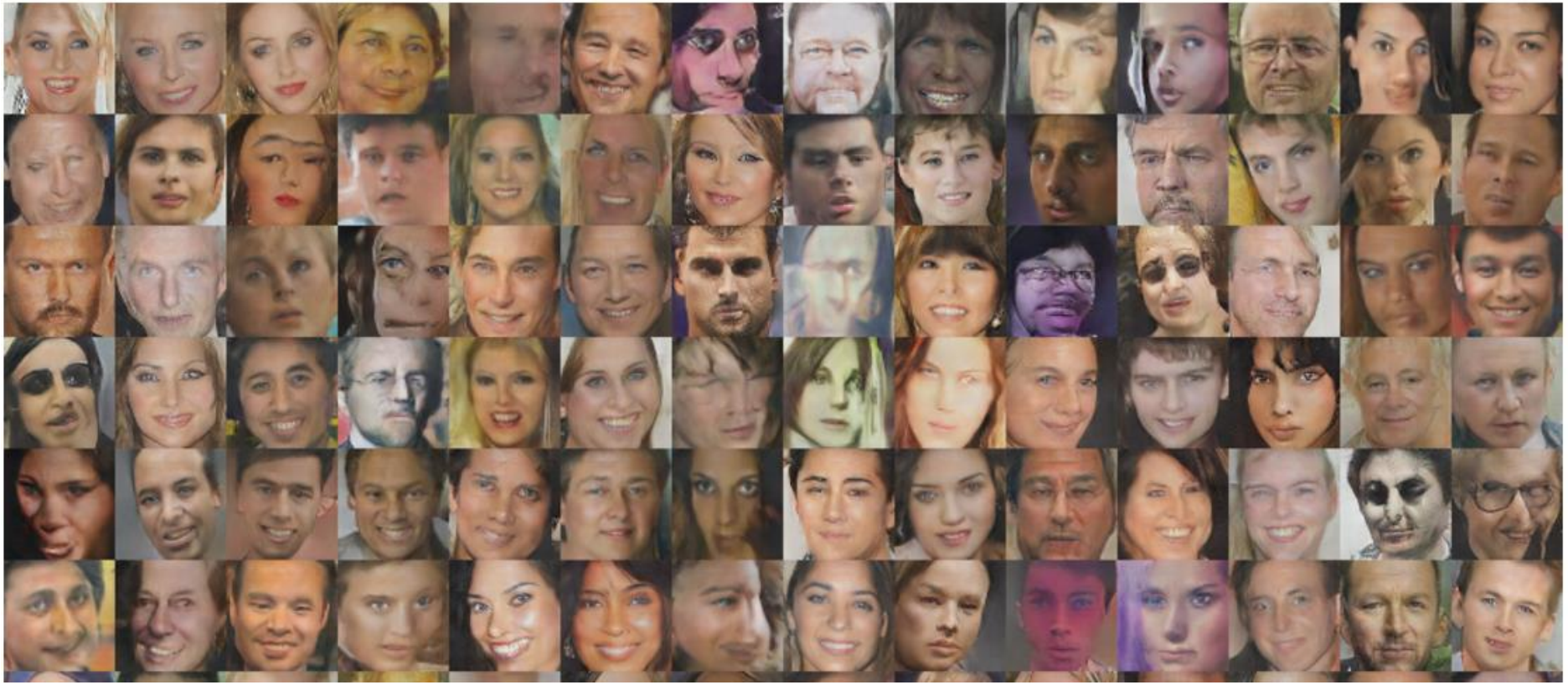
Architecture guidelines for stable Deep Convolutional GANs

- 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.

A.Radford, L. Metz, S. Chintala UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS. ICLR 2016

Примеры - спальни

Примеры - лица

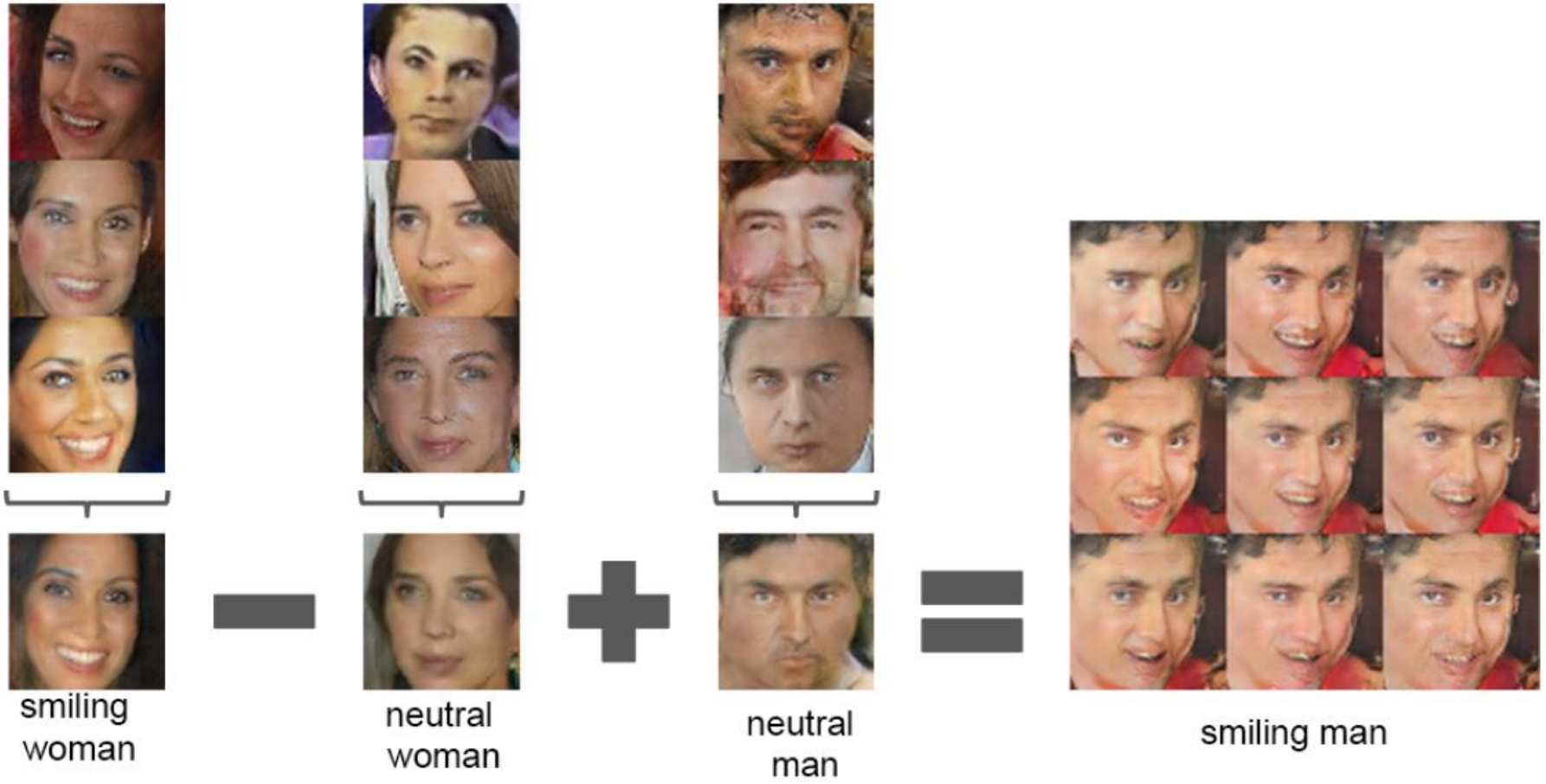


Примеры – обложки альбомов

Изучение представления

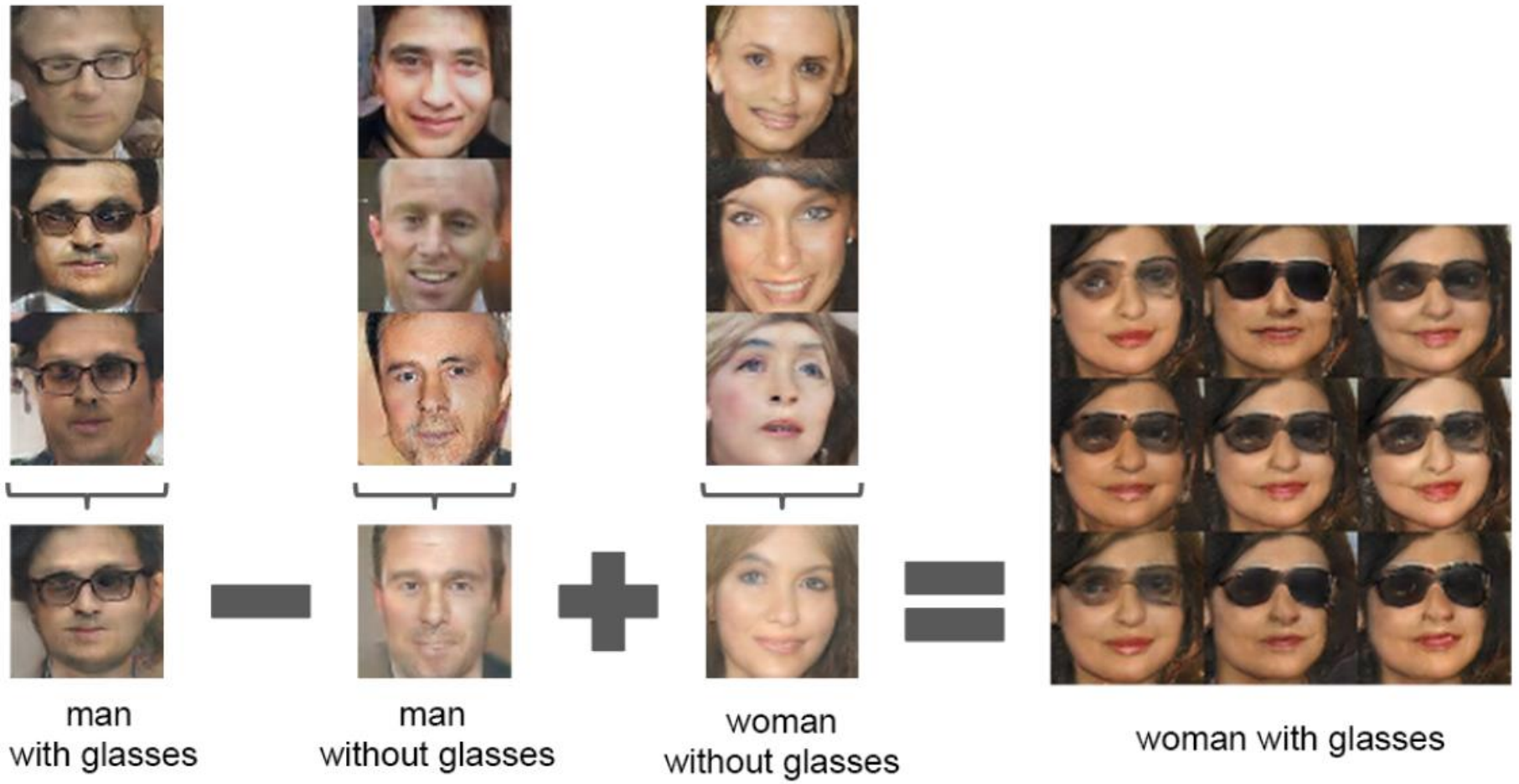
- Поиск фильтров, которые соответствуют определённым объектам
  - Логистическая регрессия по активациям фильтров внутри нарисованных областей
- Обнуление фильтров, которые были выбраны как соответствующие объектам
- На изображениях «исчезли» окна при некотором ухудшении качества
- Мораль – получаем представление, в котором за разные объекты отвечают разные фильтры

Векторная арифметика



- Работа с z-векторами, которые использовались для генерации изображений
- Отдельные вектора нестабильны, поэтому сумма по 3м изображениям

Векторная арифметика



Векторная арифметика

- Вектор «поворота» как разница между усреднёнными векторами 4х изображений с лицами повернутыми налево и 4мя направо

Laplacian GAN

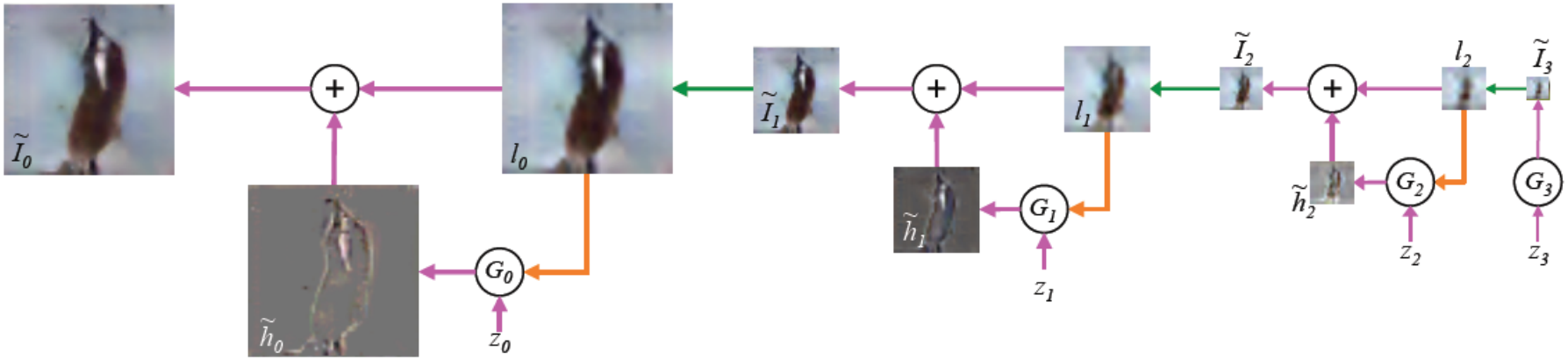


Figure 1: The sampling procedure for our LAPGAN model. We start with a noise sample z_3 (right side) and use a generative model G_3 to generate $\tilde{I}_3$. This is upsampled (green arrow) and then used as the conditioning variable (orange arrow) l_2 for the generative model at the next level, G_2. Together with another noise sample z_2, G_2 generates a difference image $\tilde{h}_2$ which is added to l_2 to create $\tilde{I}_2$. This process repeats across two subsequent levels to yield a final full resolution sample I_0.

Laplacian GAN – обучение

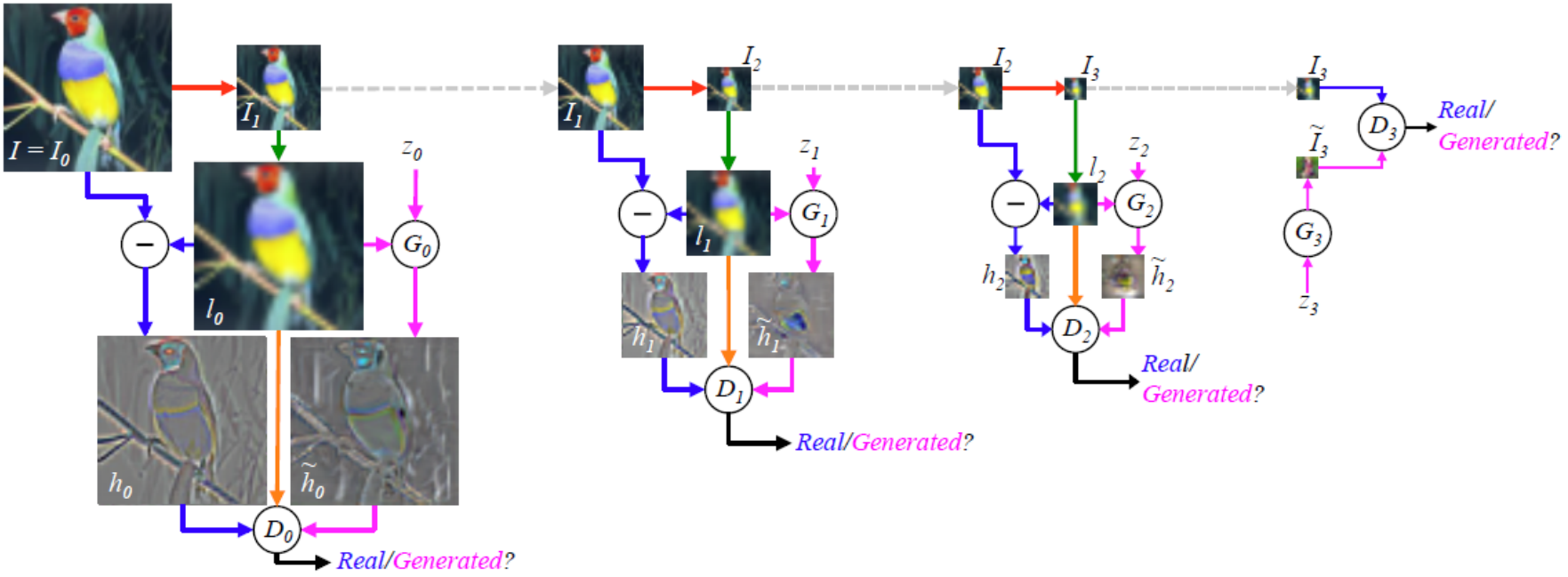
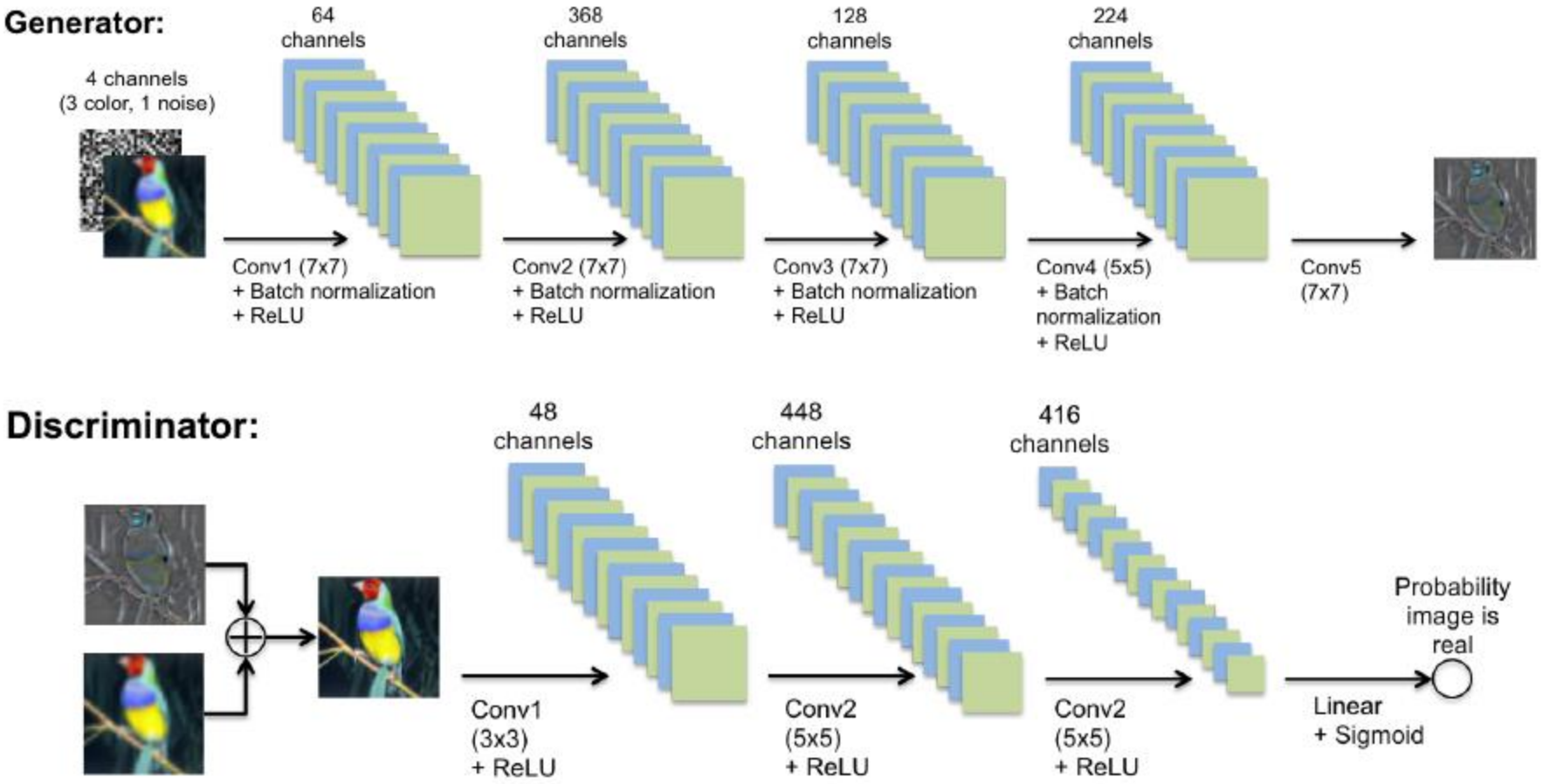


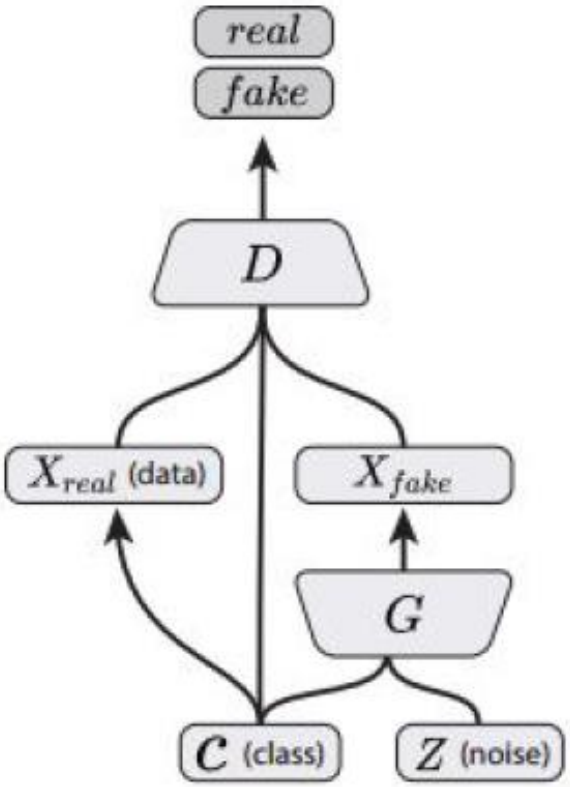
Figure 2: The training procedure for our LAPGAN model. Starting with a 64x64 input image I from our training set (top left): (i) we take $I_0 = I$ and blur and downsample it by a factor of two (red arrow) to produce I_1; (ii) we upsample I_1 by a factor of two (green arrow), giving a low-pass version l_0 of I_0; (iii) with equal probability we use l_0 to create *either* a real *or* a generated example for the discriminative model D_0. In the real case (blue arrows), we compute high-pass $h_0 = I_0 - l_0$ which is input to D_0 that computes the probability of it being real vs generated. In the generated case (magenta arrows), the generative network G_0 receives as input a random noise vector z_0 and l_0. It outputs a generated high-pass image $\tilde{h}_0 = G_0(z_0, l_0)$, which is input to D_0. In both the real/generated cases, D_0 also receives l_0 (orange arrow). Optimizing Eqn. 2, G_0 thus learns to generate realistic high-frequency structure $\tilde{h}_0$ consistent with the low-pass image l_0. The same procedure is repeated at scales 1 and 2, using I_1 and I_2. Note that the models at each level are trained independently. At level 3, I_3 is an 8×8 image, simple enough to be modeled directly with a standard GANs G_3 & D_3.

Архитектура для многих классов

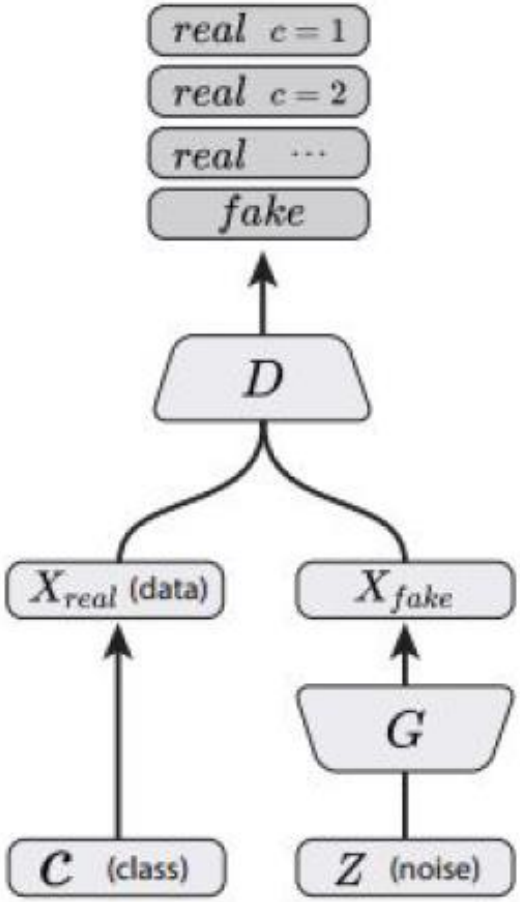


Демонстрация работы

GAN с метками



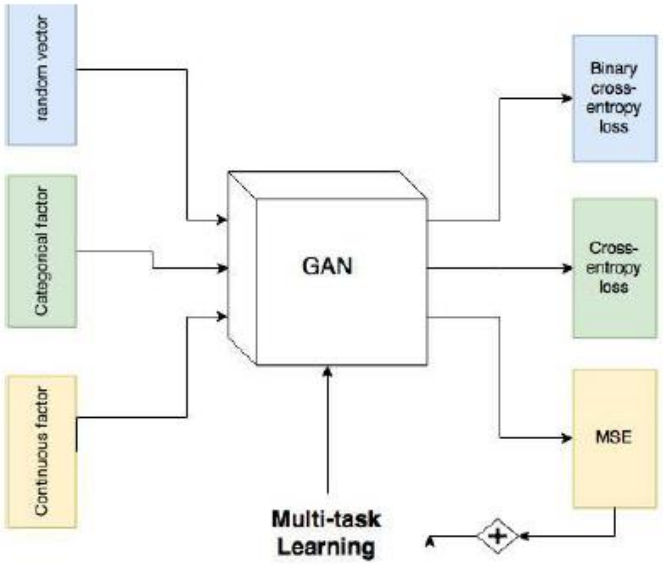
Conditional GAN
(Mirza & Osindero, 2014)



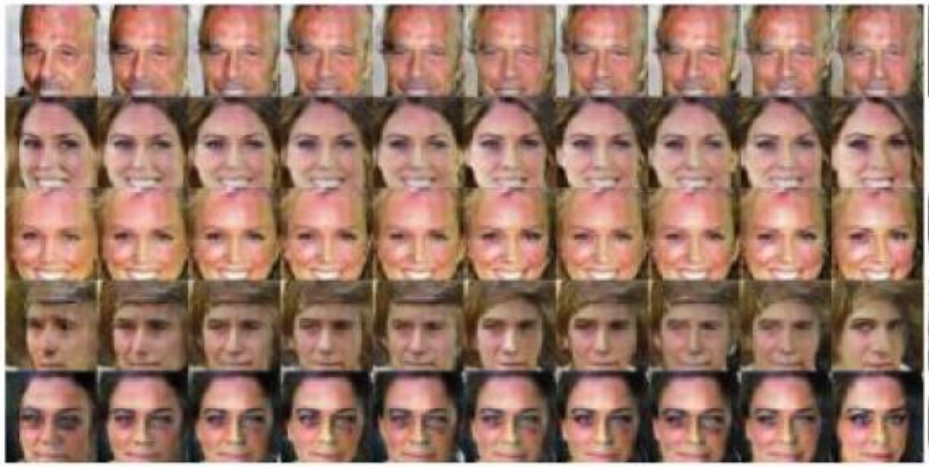
Semi-Supervised GAN
(Odena, 2016; Salimans, et al., 2016)

InfoGAN

Chen et al, InfoGAN : Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, 2016



Результаты



(a) Azimuth (pose)

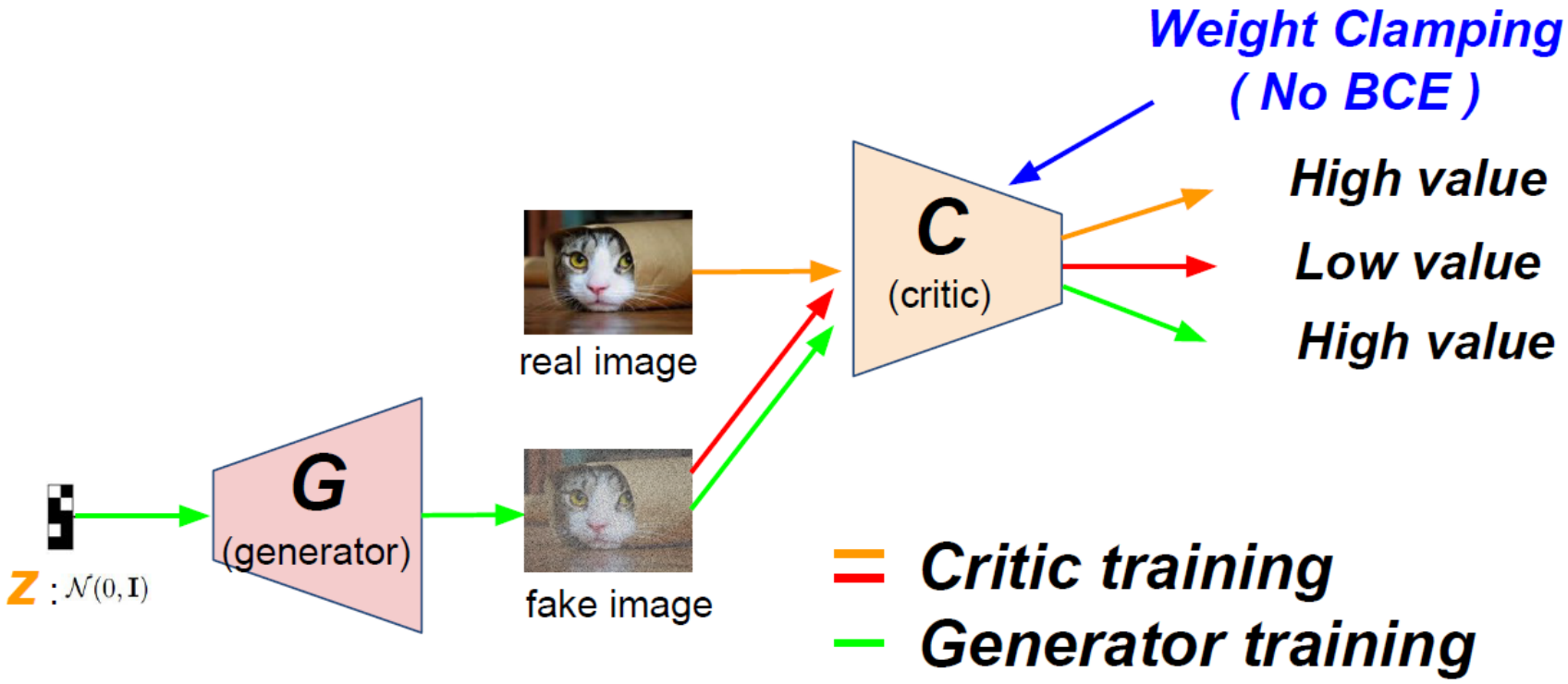
(b) Presence or absence of glasses

(c) Hair style

(d) Emotion

WGAN

- Martin et al, Wasserstein GAN, 2017



WGAN

Martin et al, Wasserstein GAN, 2017

- JSD → Earth Mover Distance(=Wasserstein-1 distance)
- Prevent the gradient vanishing by using weak distance metrics
- Provide the parsimonious training indicator.

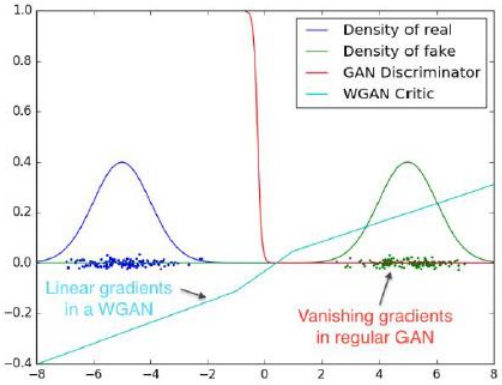
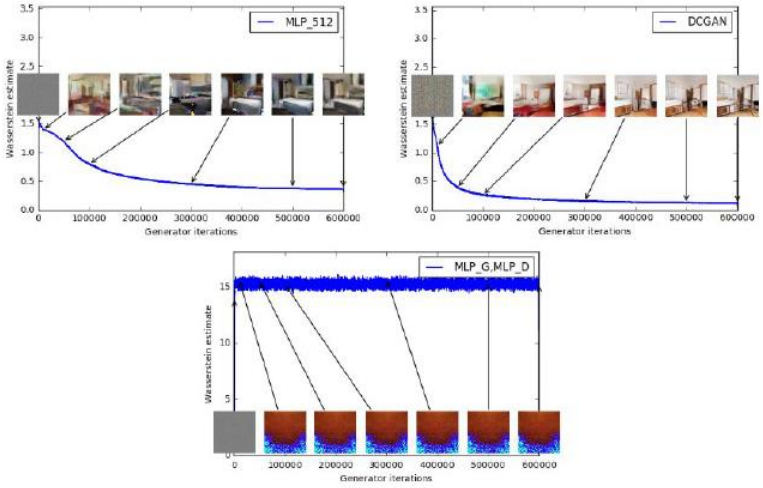
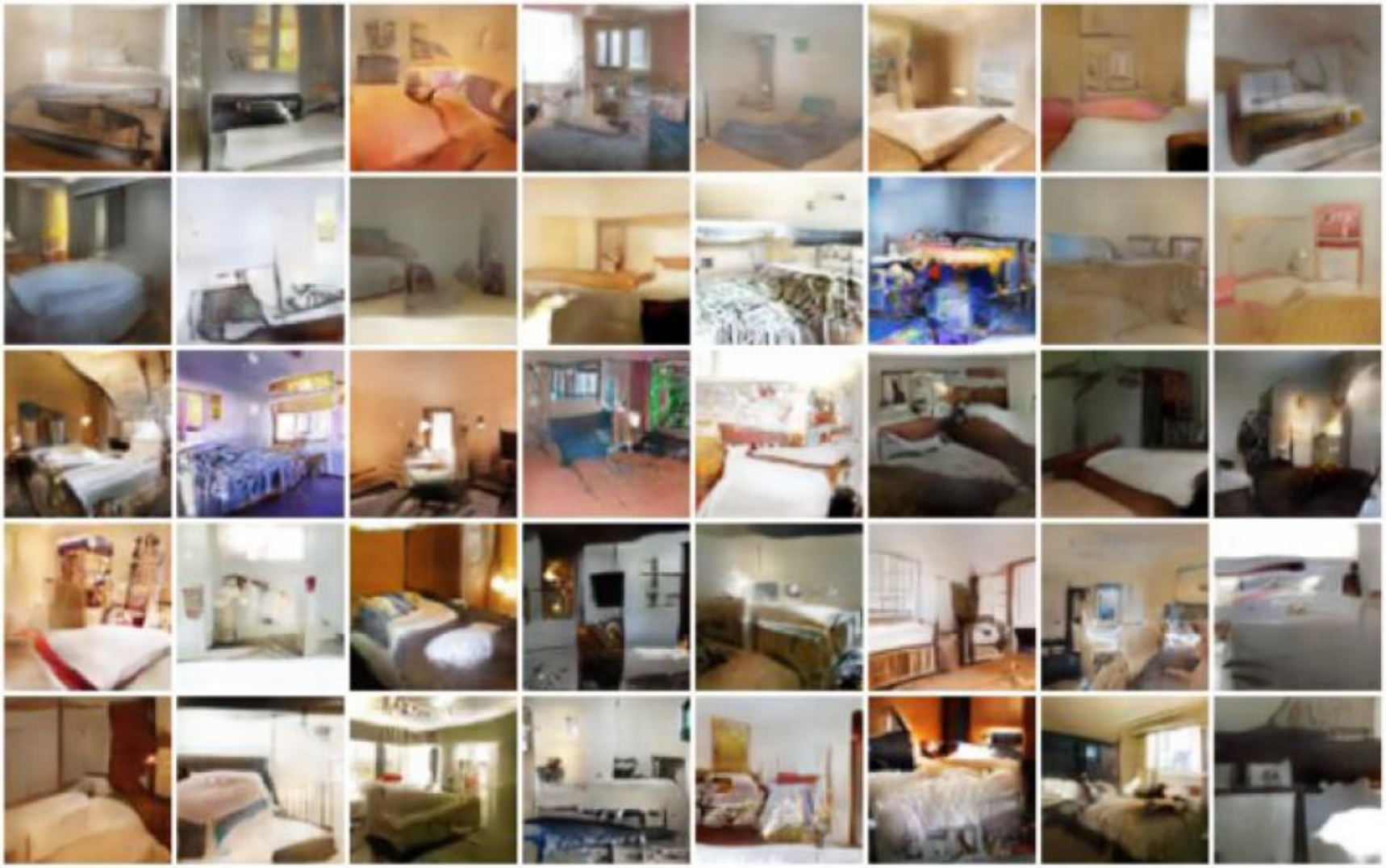


Figure 2: Optimal discriminator and critic when learning to differentiate two Gaussians. As we can see, the traditional GAN discriminator saturates and results in vanishing gradients. Our WGAN critic provides very clean gradients on all parts of the space.

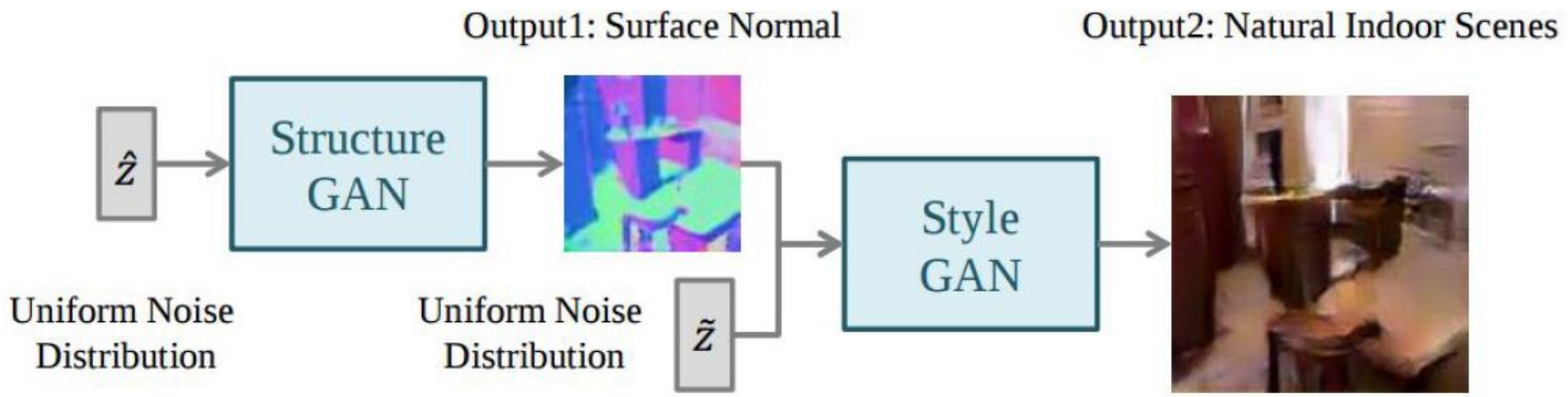


Результаты

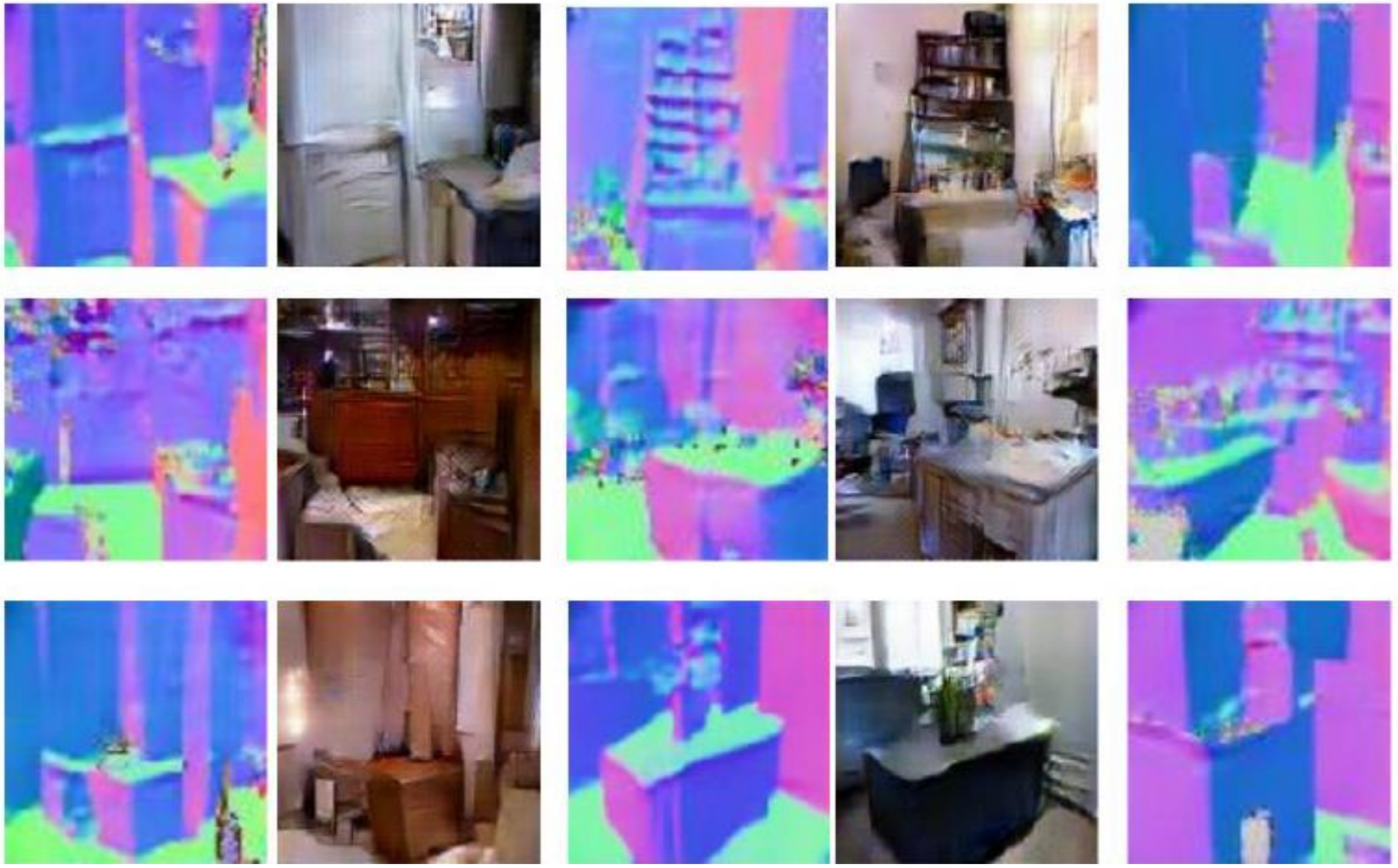


Приложения

Wang et al, Generative Image Modeling using Style and Structure Adversarial Networks, 2016

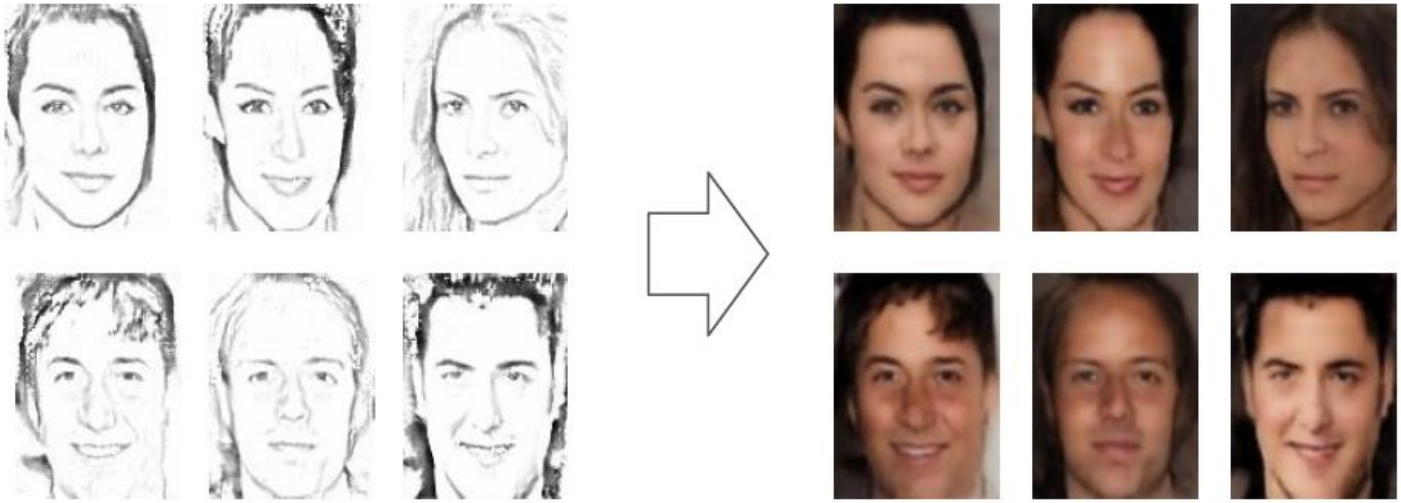


Результаты



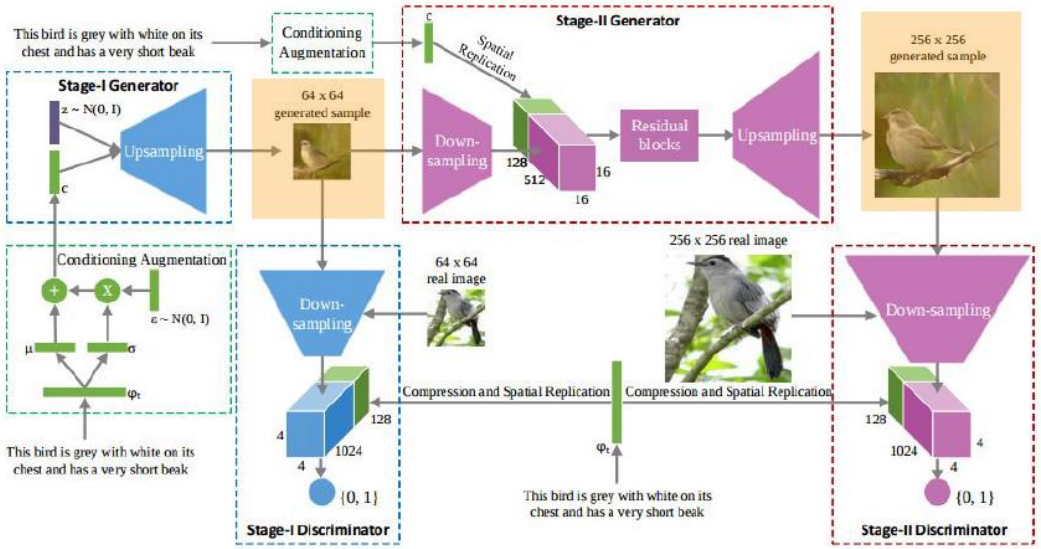
Генерация лиц

- Two-step generation : Sketch → Color
- Binomial random seed instead of gaussian



Text 2 photo

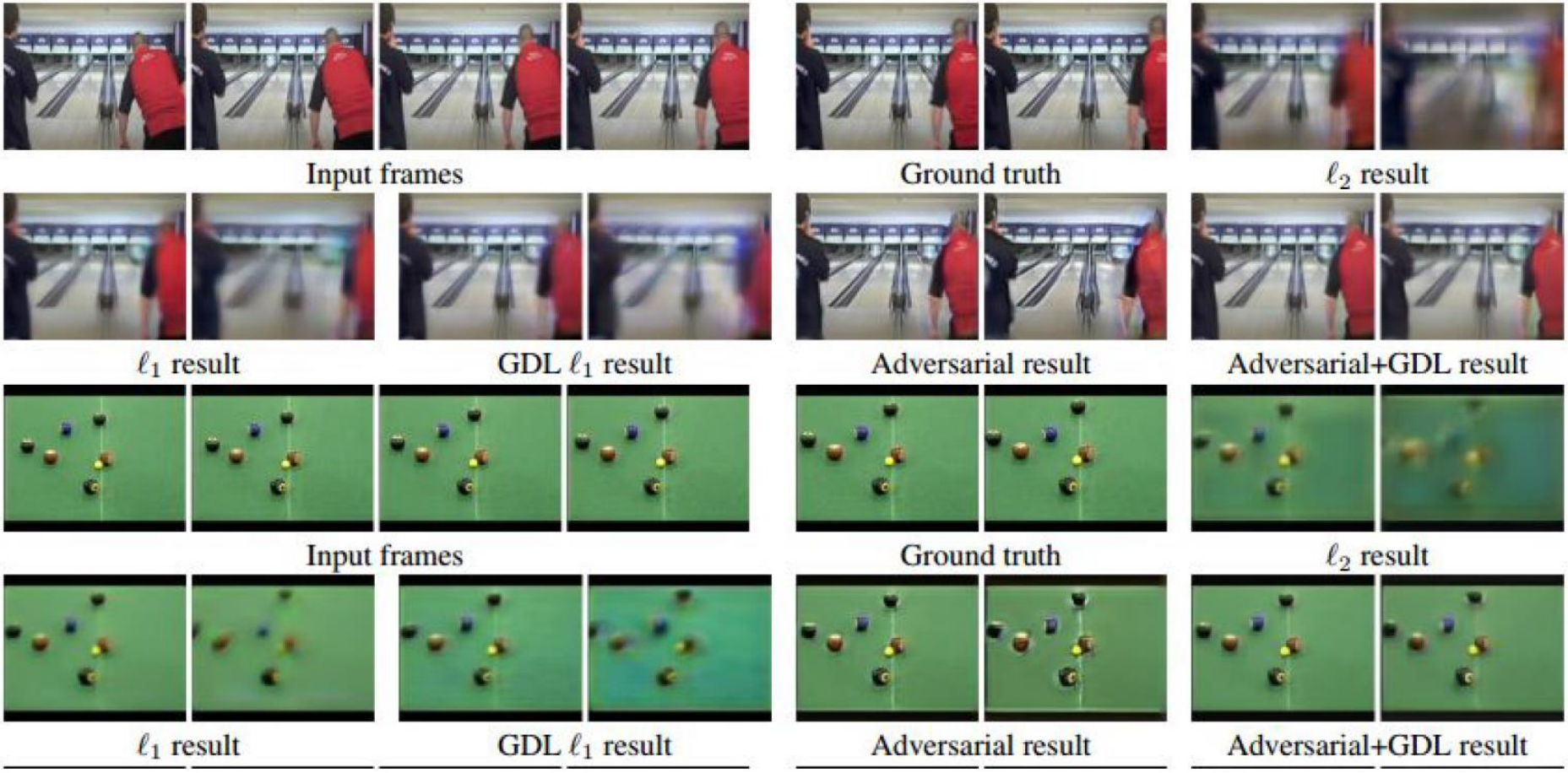
Zhang et al, StackGAN : Text to Photo-realistic Image Synthesis with Stacked GANs, 2016



Результаты

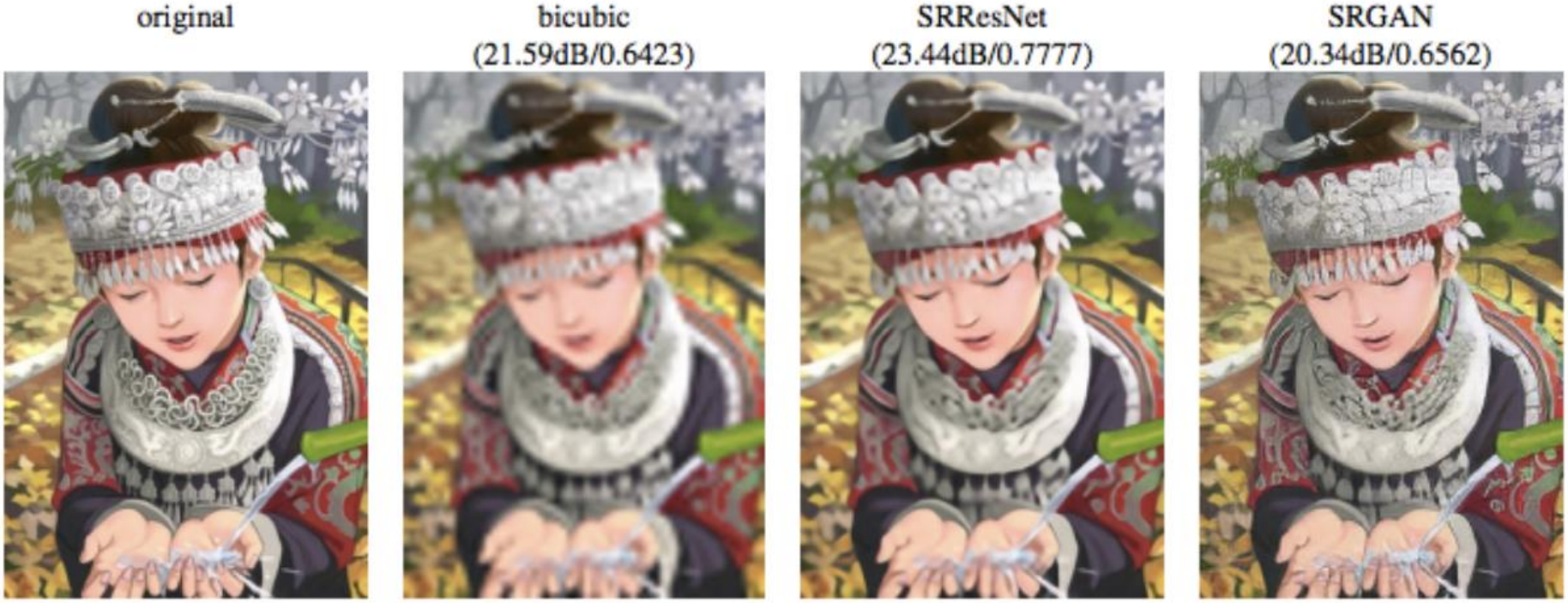

Text description
This bird is blue with white and has a very short beak
This bird has wings that are brown and has a yellow belly
A white bird with a black crown and yellow beak
This bird is white, black, and brown in color, with a brown beak
The bird has small beak, with reddish brown crown and gray belly
This is a small, black bird with a white breast and white on the wingbars.
This bird is white black and yellow in color, with a short black beak
Stage-I images
Stage-II images

Предсказание кадров



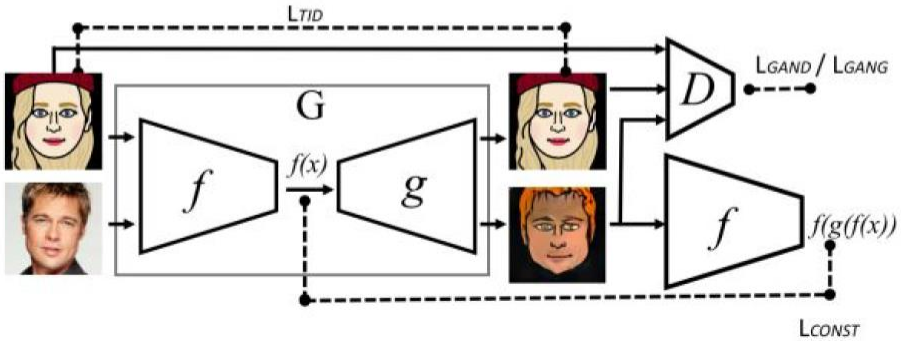
Mathieu et al. **Deep multi-scale video prediction beyond mean square error**, ICLR 2016 (https://arxiv.org/abs/1511.05440)

Суперразрешение

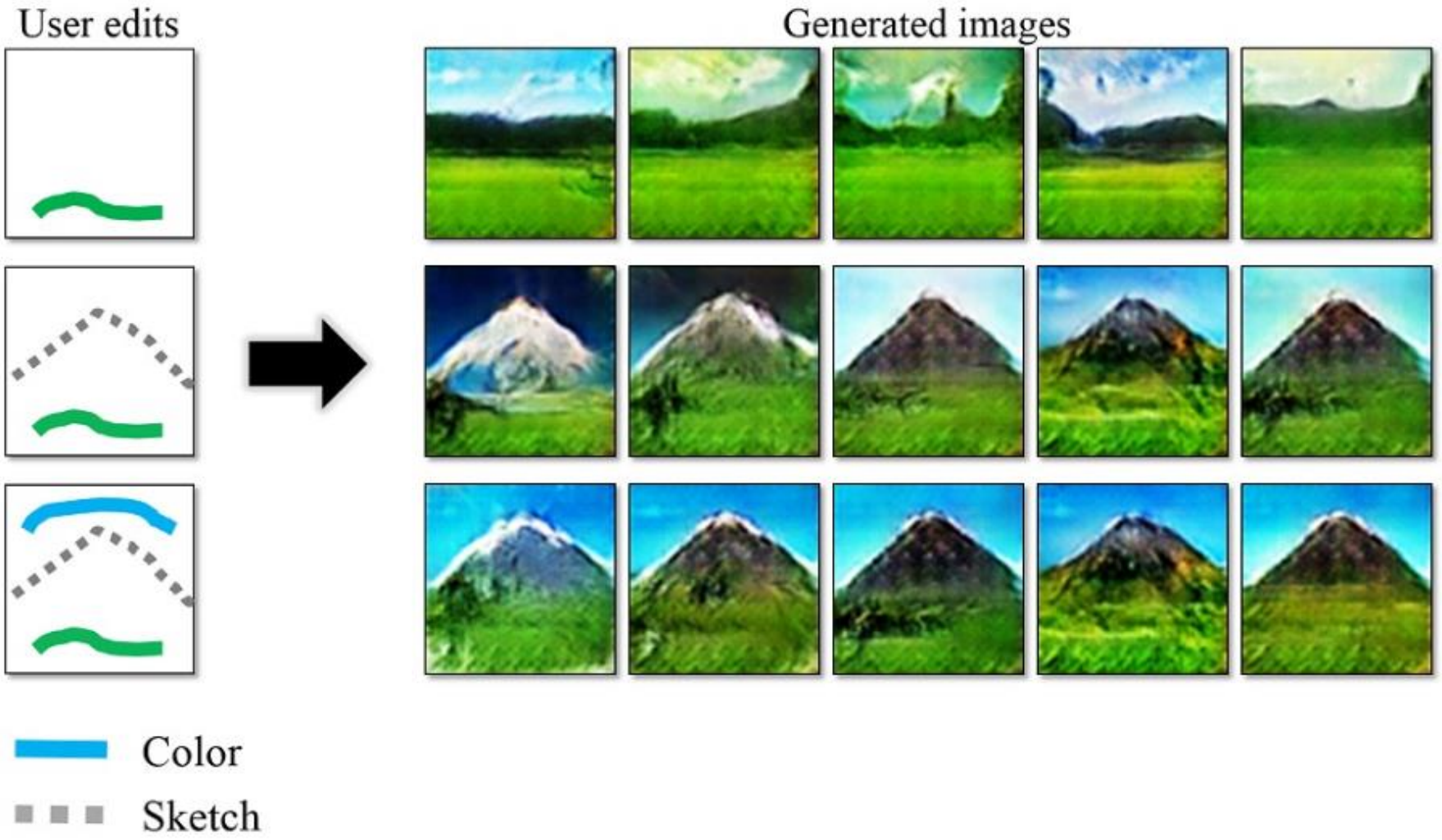


Domain Adaptation

Taigman et al, Unsupervised cross-domain image generation, 2016

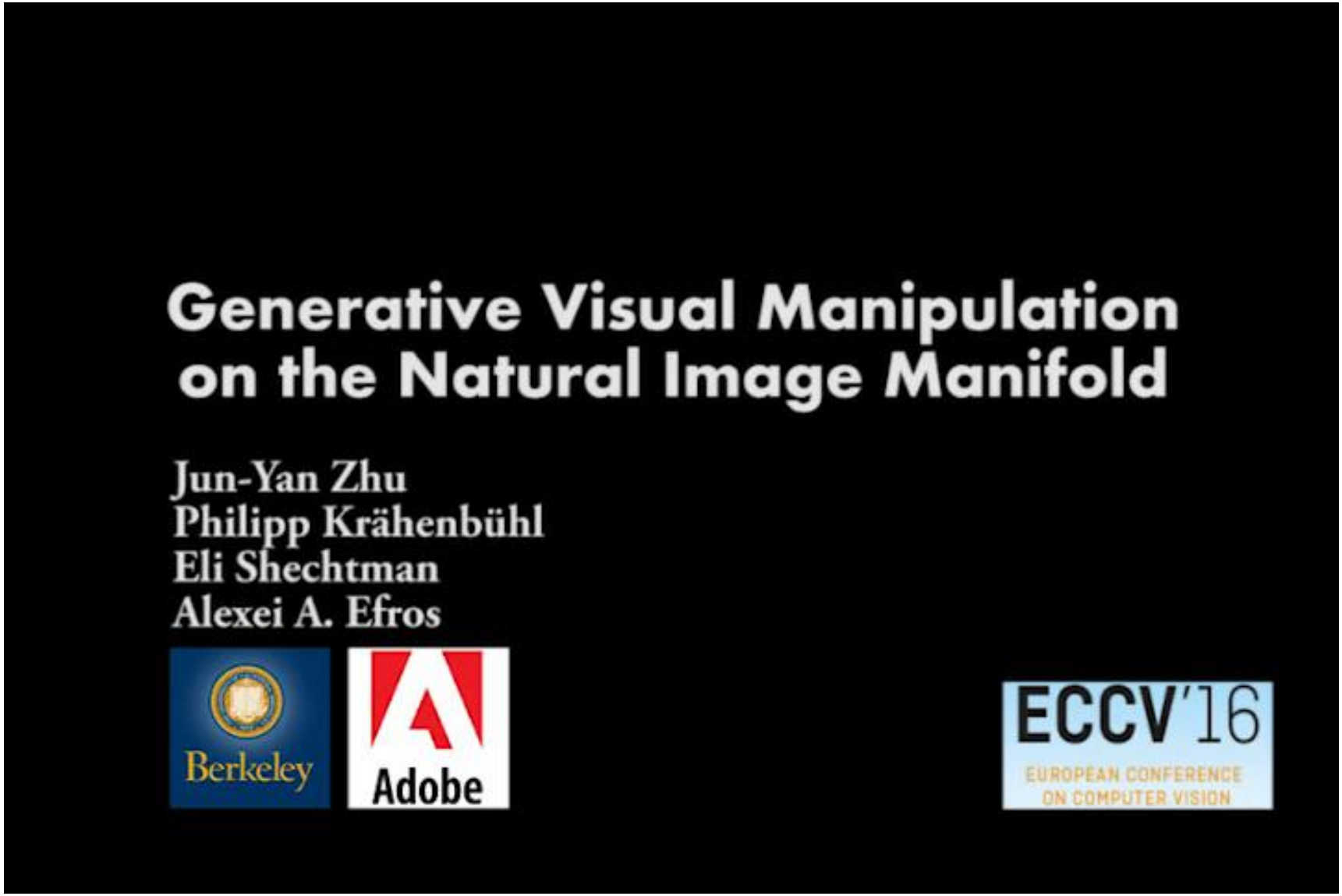


iGAN



Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman and Alexei A. Efros. "Generative Visual Manipulation on the Natural Image Manifold", in European Conference on Computer Vision (**ECCV**). 2016.

Видеопример



Резюме

GAN – одна из крутейших идей в нейросетях