VOLGOGRAD STATE TECHNICAL UNIVERSITY

Машинное обучение и нейросетевые модели

Лекция 10. Современные архитектуры GAN

Лектор: Кравченя Павел Дмитриевич

Волгоград 2025

VOLGOGRAD STATE TECHNICAL UNIVERSITY

План лекции

1. Проблемы с KL и IS. Расстояние Вассерштейна и его особенности.
2. Wasserstein GAN и принцип его работы. Обучение Wasserstein GAN.
3. Спектральная нормализация весов GAN.
4. Задачи преобразования изображений, подходы к их решению.
5. Использование perceptual loss для контроля схожести признаков и стиля.
6. Super-Resolution GAN. Архитектура генератора и дискриминатора.
7. Архитектура CycleGAN и принцип её работы. Свойство постоянства цикла.
8. Обучение CycleGAN. Виды ошибок при обучении. Классическая реализация.
9. Выделение высокоуровневых признаков в изображениях. StyleGAN.
10. Style-Based генератор. Адаптивная instance-нормализация. Стили.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Проблемы метрики Йенсена-Шеннона

- Задача *обучения генеративно-состязательных сетей* сводится к попытке научить генератор G создавать данные, *распределение* которых $p_g(\mathbf{x})$ *максимально близко* к распределению *реальных* данных $p_r(\mathbf{x})$.
- Для оценки этой «*близости*» нужно уметь измерять «*расстояние*» между $p_r(\mathbf{x})$ и $p_g(\mathbf{x})$, для чего часто используются *дивергенции Кульбака-Лейблера* или *Йенсена-Шеннона*.
- Однако, у данных способов измерения степени схожести распределений *имеются проблемы*, такие как *нестабильность обучения* и *исчезающий градиент*, особенно если распределения $p_r(\mathbf{x})$ и $p_g(\mathbf{x})$ *не пересекаются*.
- Можно ли оценивать схожесть распределений *по-другому*?

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Расстояние Вассерштейна

- Еще одной мерой *схожести двух распределений* является *расстояние Вассерштейна* (Earth Mover's distance):

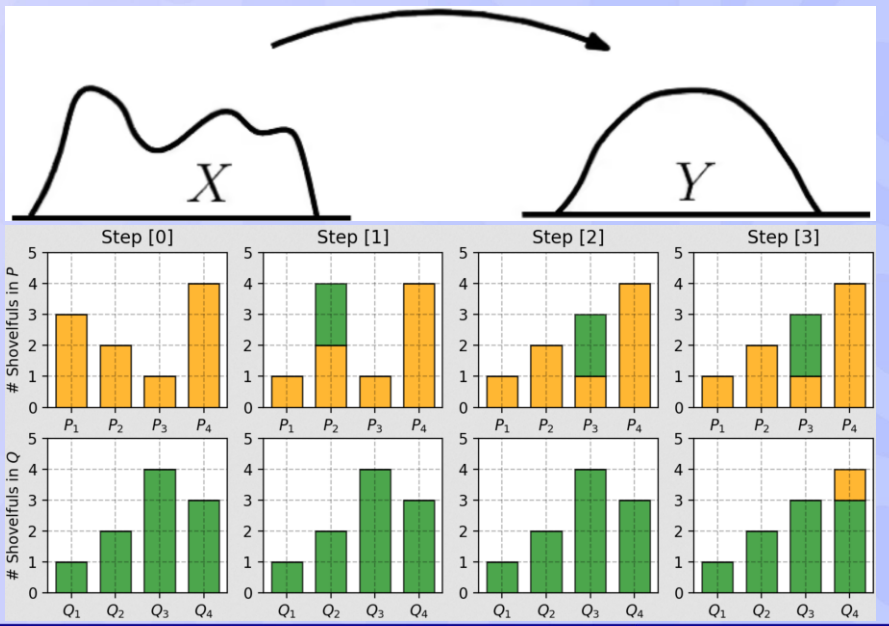
$$W(p_r, p_g) = \inf_{\gamma \sim \Pi(p_r, p_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|\mathbf{x} - \mathbf{y}\|].$$

- Расстояние Вассерштейна можно интерпретировать как «*стоимость*» перемещения земли между двумя кучами.
- Рассмотрим пример *дискретного* случая.

Пусть $\delta_{i+1} = \delta_i + P_i - Q_i$. Тогда:

$$\delta_0 = 0, \quad \delta_1 = 0 + 3 - 1 = 2, \quad \delta_2 = 2 + 2 - 2 = 2,$$
$$\delta_3 = 2 + 1 - 4 = -1, \qquad \delta_4 = -1 + 4 - 3 = 0.$$

- Тогда *расстояние* $W = \sum_i |\delta_i| = 5$.

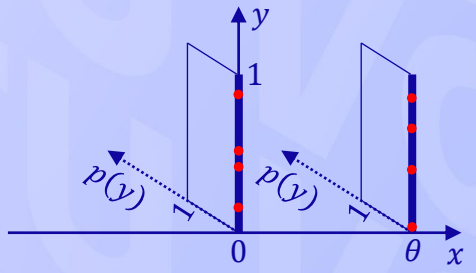


VOLGOGRAD STATE TECHNICAL UNIVERSITY

Пример расчета различных метрик схожести распределений

- Рассмотрим *два распределения* $p(x,y)$ и $q(x,y)$ таких, что:

$$\forall (x,y) \in p(x,y) \hookrightarrow x = 0, y \sim \mathcal{U}(0,1),$$
$$\forall (x,y) \in q(x,y) \hookrightarrow x = \theta, y \sim \mathcal{U}(0,1).$$



- Когда $\theta \neq 0$:

$$\mathrm{KL}(p||q) = \sum_{\substack{x=0 \\ y\sim\mathcal{U}(0,1)}} p(x,y) \cdot \log\frac{p(x,y)}{q(x,y)} = \sum_{\substack{x=0 \\ y\sim\mathcal{U}(0,1)}} 1 \cdot \log\frac{1}{0} = +\infty,$$

$$\mathrm{KL}(q||p) = \sum_{\substack{x=\theta \\ y\sim\mathcal{U}(0,1)}} q(x,y) \cdot \log\frac{q(x,y)}{p(x,y)} = \sum_{\substack{x=0 \\ y\sim\mathcal{U}(0,1)}} 1 \cdot \log\frac{1}{0} = +\infty,$$

$$\mathrm{JS}(p||q) = \frac{1}{2}\left[\sum_{\substack{x=0 \\ y\sim\mathcal{U}(0,1)}} p(x,y) \cdot \log\frac{2 \cdot p(x,y)}{p(x,y)+q(x,y)} + \sum_{\substack{x=\theta \\ y\sim\mathcal{U}(0,1)}} q(x,y) \cdot \log\frac{2 \cdot q(x,y)}{p(x,y)+q(x,y)}\right] = \log 2\,,$$

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Преимущество расстояния Вассерштейна для GAN

$$W(p, q) = |\theta|.$$

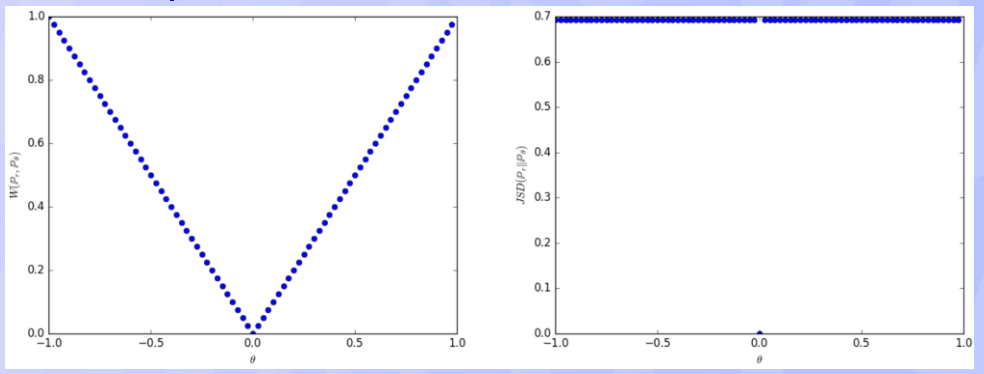
- Но когда $\theta = 0$, распределения $p(x, y)$ и $q(x, y)$ *полностью перекрываются*.

Тогда:

$$\mathrm{KL}(p||q) = \mathrm{KL}(q||p) = \mathrm{JS}(p||q) = 0,$$
$$W(p, q) = 0 = |\theta|.$$

- Видно, что если распределения не совпадают, дивергенция Кульбака-Лейблера равна *бесконечности*. Значение дивергенции Йенсена-Шеннона испытывает *скачок*, что приводит к её *недифференцируемости* в точке $\theta = 0$.

А расстояние Вассерштейна остаётся *гладким*, что обеспечивает *стабильный процесс обучения* GAN.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

Соотношение двойственности Канторовича-Рубенштейна

- В выражении для расчета расстояния Вассерштейна присутствует *infimum* по *распределениям*, расчет которого вычислительно очень сложен (*intractable*). Поэтому, пользуются соотношением двойственности *Канторовича-Рубинштейна*:

$$W(p_r, p_g) = \frac{1}{K} \sup_{\|f\|_L \leq K} \left[\mathbb{E}_{x \sim p_r}[f(\mathbf{x})] - \mathbb{E}_{x \sim p_g}[f(\mathbf{x})] \right].$$

Здесь $f\colon \mathcal{X} \to \mathbb{R}$ – *K-Lipschitz continuous* функция, т.е. для которой:

$$\exists K \geq 0 : \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X} \hookrightarrow |f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq K|\mathbf{x}_1 - \mathbf{x}_2|.$$

- Предполагая, что $f = f_w$ является *K-Lipschitz continuous* и рассчитывается с помощью *дискриминатора* в GAN, получим *Wasserstein Loss* для GAN:

$$\mathcal{L}(p_r, p_g) = W(p_r, p_g) = \max_{\mathbf{w} \in \mathcal{W}} \left[\mathbb{E}_{\mathbf{x} \sim p_r}[f_{\mathbf{w}}(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim p_z}[f_{\mathbf{w}}(g_{\boldsymbol{\theta}}(\mathbf{z}))] \right].$$

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Алгоритм обучения WGAN

while θ has not converged **do**

  for $t = 0, \ldots, n_{critic}$ **do**

    Sample $\{\mathbf{x}^{(i)}\}_{i=1}^{m} \sim p_r(\mathbf{x})$ batch from the real data;

    Sample $\{\mathbf{z}^{(i)}\}_{i=1}^{m} \sim p_z(\mathbf{z})$ batch of prior samples;

    $\mathbf{w} \leftarrow \mathbf{w} + \eta \cdot \mathrm{RMSProp}\left(\mathbf{w}, \nabla_{\mathbf{w}}\left[\frac{1}{m}\sum_{i=1}^{m} f_{\mathbf{w}}(\mathbf{x}^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_{\mathbf{w}}\left(g_{\boldsymbol{\theta}}(\mathbf{z}^{(i)})\right)\right]\right);$

    $\mathbf{w} \leftarrow \mathrm{clip}(\mathbf{w}, -c, c);$ # **Weight Clipping**

  end for

  Sample $\{\mathbf{z}^{(i)}\}_{i=1}^{m} \sim p_z(\mathbf{z})$ batch of prior samples;

  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \cdot \mathrm{RMSProp}\left(\boldsymbol{\theta}, \nabla_{\boldsymbol{\theta}}\left[\frac{1}{m}\sum_{i=1}^{m} f_{\mathbf{w}}\left(g_{\boldsymbol{\theta}}(\mathbf{z}^{(i)})\right)\right]\right);$

end while

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Особенности обучения WGAN

- Для того, чтобы функция *дискриминатора* удовлетворяла условию *K-Lipschitz continuous*, *веса* дискриминатора *ограничивают* в процессе обучения.
- Модель дискриминатора выступает не в качестве *прямого критика*, а в качестве *помощника* при оценке метрики Вассерштейна между *реальным* и *сгенерированным* распределениями данных.
- Авторы алгоритма рекомендовали использовать *RMSProp*, а не *Adam*, поскольку он мог вызвать *нестабильность* в обучении модели.
- WGAN не является *совершенным*; он страдает от *нестабильного обучения* и *медленной сходимости* после ограничения весов.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

- Для того, чтобы для обучения критика *можно было применить* условие *Канторовича-Рубенштейна*, *функция критика* должна быть *Липшицевой*.
- По определению, *Липшицева норма* (наименьшая константа K, при которой выполняется условие *K-Lipschitz continuous*):

$$\|f\|_{\text{Lip}} = \sup_{\mathbf{x}_1 \neq \mathbf{x}_2} \frac{\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\|_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|_2}.$$

- Если функция f дифференцируема, то: $f(\mathbf{x}_1) - f(\mathbf{x}_2) = \nabla f(\mathbf{h}) \cdot \Delta\mathbf{x}$, $\mathbf{h} \in [\mathbf{x}_1, \mathbf{x}_2]$, и:

$$\|f\|_{\text{Lip}} = \sup_{\mathbf{x}_1 \neq \mathbf{x}_2} \frac{\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\|_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|_2} = \sup_{\Delta\mathbf{x} \neq 0} \frac{\|\nabla f(\mathbf{h}) \cdot \Delta\mathbf{x}\|_2}{\|\Delta\mathbf{x}\|_2}.$$

- При *фиксированной* норме $\|\Delta\mathbf{x}\|_2$ требуется *подобрать* $\|\nabla f(\mathbf{h}) \cdot \Delta\mathbf{x}\|_2$ так, чтобы *норма Липшица была максимальной*.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Контроль константы Липшица

- *Спектральная норма* матрицы $\mathbf{A}$ определяется как:

$$\sigma(\mathbf{A}) = \max_{\|\mathbf{x}\|_2} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}.$$

- Таким образом, *норма Липшица*:

$$\|f\|_{\text{Lip}} = \sup_{\Delta\mathbf{x}\neq 0} \frac{\|\nabla f(\mathbf{h}) \cdot \Delta\mathbf{x}\|_2}{\|\Delta\mathbf{x}\|_2} = \sup_{\mathbf{h}} \sigma\big(\nabla f(\mathbf{h})\big).$$

- Для *линейного слоя* $f(\mathbf{h}) = \mathbf{W}\mathbf{h}$ в нейронной сети:

$$\|f\|_{\text{Lip}} = \sup_{\mathbf{h}} \sigma\big(\nabla(\mathbf{W}\mathbf{h})\big) = \sup_{\mathbf{h}} \sigma(\mathbf{W}) = \sigma(\mathbf{W}).$$

- Пусть *Липшицева норма* функции активации $\|a_l\|_{\text{Lip}}$ равна *единице* (это справедливо для *ReLU*, *LeakyReLU* и других *популярных функций*): $\|a_l\|_{\text{Lip}} = 1$.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Контроль константы Липшица

- *Можно показать* справедливость неравенства:

$$\|f_1 \circ f_2\|_{\mathrm{Lip}} \leq \|f_1\|_{\mathrm{Lip}} \cdot \|f_2\|_{\mathrm{Lip}}.$$

- Применяя его к *многослойной полносвязной нейросети*, оценим её *норму Липшица*:

$$\|f\|_{\mathrm{Lip}} \leq \|\mathbf{W}^{L+1}\mathbf{h}_L\|_{\mathrm{Lip}} \cdot \|a_L\|_{\mathrm{Lip}} \cdot \|\mathbf{W}^{L}\mathbf{h}_{L-1}\|_{\mathrm{Lip}} \cdots \|a_1\|_{\mathrm{Lip}} \cdot \|\mathbf{W}^{1}\mathbf{h}_0\|_{\mathrm{Lip}} = \prod_{l=1}^{L+1} \sigma(\mathbf{W}^l).$$

- Если *нормализовать спектральную норму матриц весов*:

$$\bar{\mathbf{W}}_{\mathrm{SN}}(\mathbf{W}) = \frac{\mathbf{W}}{\sigma(\mathbf{W})},$$

то $\sigma\left(\bar{\mathbf{W}}_{\mathrm{SN}}(\mathbf{W})\right) = 1$, и *норма Липшица нейросети*: $\boxed{\|f\|_{\mathrm{Lip}} \leq 1.}$

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Спектральная нормализация

- **Спектральная нормализация** – это метод *регуляризации*, используемый в генеративно-состязательных сетях (GAN) для *стабилизации обучения* и *улучшения качества* генерируемых данных.
- Обычно она применяется к весам *дискриминатора* (*критика*) с целью *ограничения* его «мощности» и *предотвращения* таких *проблем*, как исчезающий градиент или нестабильность обучения.
- Для матрицы весов $\mathbf{W}$ в слое нейронной сети *спектральная норма*:

$$\sigma(\mathbf{W}) = \max_{\mathbf{x} \neq 0} \frac{\|\mathbf{W}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}.$$

Здесь $\|\cdot\|_2$ – *евклидова норма*. Спектральная нормализация *ограничивает спектральную норму* матрицы весов каждого слоя нейронной сети.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Алгоритм SGD со спектральной нормализацией

Инициализировать $\tilde{\mathbf{u}}_l \sim \mathcal{N}(\mathbf{0}, \sigma\mathbb{I})$ для всех слоёв $l \in [1..L]$.

for $t = 1, \ldots, T$ **do**

  for $l = 1, \ldots, L$ **do**

    Применить итерационный метод к ненормализованной матрице:

$$\tilde{\mathbf{v}}_l \leftarrow \frac{(\mathbf{W}^l)^{\mathrm{T}} \tilde{\mathbf{u}}_l}{\|(\mathbf{W}^l)^{\mathrm{T}} \tilde{\mathbf{u}}_l\|_2}, \qquad \tilde{\mathbf{u}}_l \leftarrow \frac{\mathbf{W}^l \tilde{\mathbf{v}}_l}{\|\mathbf{W}^l \tilde{\mathbf{v}}_l\|_2}.$$

    Вычислить спектральную нормализацию:

$$\bar{\mathbf{W}}^l{}_{\mathrm{SN}}(\mathbf{W}^l) = \frac{\mathbf{W}^l}{\sigma(\mathbf{W}^l)}, \qquad \sigma(\mathbf{W}^l) = \tilde{\mathbf{u}}_l^{\mathrm{T}} \mathbf{W}^l \tilde{\mathbf{v}}_l.$$

    Обновить веса на мини-батче $\mathcal{D}_M$:

$$\mathbf{W}^l \leftarrow \mathbf{W}^l - \eta \nabla_{\mathbf{W}^l} \mathcal{L}(\bar{\mathbf{W}}^l{}_{\mathrm{SN}}(\mathbf{W}^l), \mathcal{D}_M).$$

  end for

end for

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Обобщение функций ошибок GAN

Название	Формула	Особенности / свойства
Saturation Loss (классическая функция ошибки GAN)	$\min_G \max_D \left[\mathbb{E}_{p_r(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{p_z(\mathbf{z})}\left[\log\left(1 - D\left(G(\mathbf{z})\right)\right)\right]\right]$	• Может приводить к проблеме *исчезающего градиента*, особенно на ранних стадиях обучения. • Генератор «*насыщается*», если дискриминатор *слишком сильный*.
Non-Saturation Loss (неклассическая функция ошибки GAN)	$\min_G \max_D \left[\mathbb{E}_{p_r(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{p_z(\mathbf{z})}\left[-\log D\left(G(\mathbf{z})\right)\right]\right]$	• Улучшает обучение генератора, так как градиенты для генератора *не исчезают*. • *Максимизирует вероятность*, что созданные данные классифицируются как реальные. • Часто используется на практике для *стабилизации обучения*.
Wasserstein Loss (функция ошибки для WGAN)	$\min_G \max_D \left[\mathbb{E}_{p_r(\mathbf{x})}[D(\mathbf{x})] - \mathbb{E}_{p_z(\mathbf{z})}\left[D\left(G(\mathbf{z})\right)\right]\right]$	• Основана на *расстоянии Вассерштейна*, улучшает *стабильность обучения*. • Требует, чтобы дискриминатор был *1-Липшицевым*. • *Уменьшает проблемы* с исчезающим градиентом и коллапсом моды.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Обобщение способов регуляризации дискриминатора GAN

Название	Формула	Особенности / свойства
Weight Clipping (Original WGAN)	$w_i \leftarrow \text{clip}(w_i, -c, c), \quad w_i \in \mathbf{W}$	• Простой способ *ограничения весов*. Накладывает *ограничение Липшица* на дискриминатор. • Может привести к *нестабильности обучения*, если значение c выбрано *неудачно*.
Gradient Penalty (градиентный штраф)	$\lambda \cdot \mathbb{E}_{p_{\hat{x}}(\hat{\mathbf{x}})}[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2],$ $\hat{x} = \varepsilon \mathbf{x} + (1-\varepsilon)\tilde{\mathbf{x}},$ $\mathbf{x} \sim p_r(\mathbf{x}), \ \tilde{\mathbf{x}} \sim p_g(\tilde{\mathbf{x}}), \quad \varepsilon_i \sim \mathcal{U}(0,1) \ \forall i$	• Используется в WGAN-GP для обеспечения *Липшицевости критика*. • Штрафует отклонение *градиента дискриминатора* от нормы 1. • Улучшает *стабильность обучения* и *качество генерации*.
Spectral Normalization (спектральная нормализация)	$\bar{\mathbf{W}}_{\text{SN}}(\mathbf{W}) = \frac{\mathbf{W}}{\sigma(\mathbf{W})}$	• *Нормализует веса* дискриминатора для обеспечения Липшицевости. • Используется в *SN-GAN*. • *Прост в реализации, вычислительно эффективен, улучшает стабильность обучения*.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

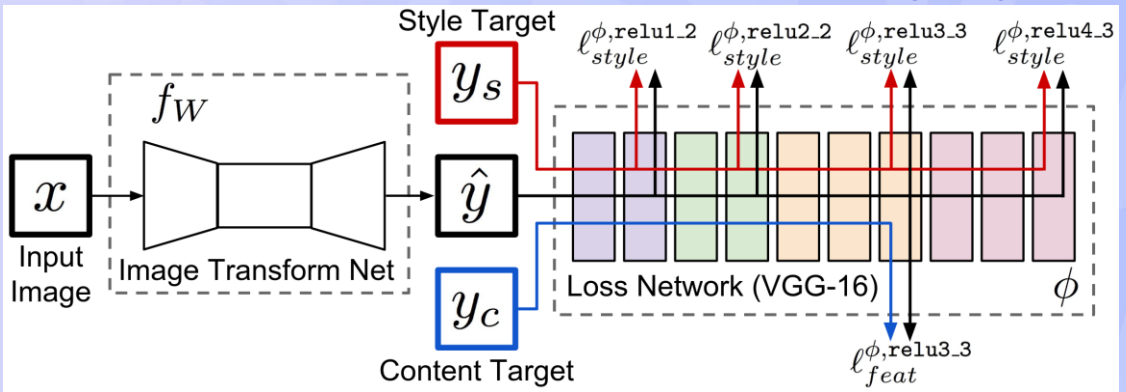
Подход к решению задач преобразования изображений

- В задачах *преобразования изображений* требуется одно изображение *преобразовать* в другое (шумоподавление, увеличение разрешения, раскраска и т.д.).
- Данные задачи обычно решаются с помощью *нейросети*, обученной с использованием *ошибки – различия* между изображениями.
- Однако, как измерять *различие между изображениями*?
- *Наивный подход*: попиксельная ошибка (*per-pixel loss*). Однако, данная функция ошибки не учитывает *внутренней структуры* изображений.
- *Улучшенный подход*: использование ошибки восприятия (*perceptual loss*). Данная функция ошибки использует *высокоуровневые признаки*, связанные с *семантикой* изображений.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Контроль схожести признаков и стиля с помощью perceptual loss

- В *SRGAN* предлагается способ контроля *схожести* изображений и стиля.
- *Идея*: ввести две нейронные сети – transformation network и loss network.
  - *Первая* используется для преобразования *исходного* изображения $\mathbf{x} \in X$ в *выходное* $\hat{\mathbf{y}} \in Y$: $\hat{\mathbf{y}} = f_W(\mathbf{x})$.
  - *Вторая* осуществляет преобразование *входного* изображения в *тензор признаков*, содержащих информацию о *структуре изображения*, которые вместе используются для *сравнения* признаков изображений и стиля с применением *perceptual loss functions*.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

Функции ошибки для контроля схожести признаков и стиля

- *Loss network* используется для оценки *ошибки реконструкции признаков* (*feature reconstruction loss*) ℓ_{feat}^{ϕ} и *ошибки реконструкции стиля* (*style reconstruction loss*) ℓ_{style}^{ϕ}. Эти ошибки и составляют ошибку восприятия (*perceptual loss*). Соответствующие *функции* определяются следующим образом:

$$\ell_{feat}^{\phi,j}(\hat{\mathbf{y}},\mathbf{y}) = \frac{1}{C_j H_j W_j}\left\|\phi_j(\hat{\mathbf{y}}) - \phi_j(\mathbf{y})\right\|_2^2.$$

$$\ell_{style}^{\phi,j}(\hat{\mathbf{y}},\mathbf{y}) = \frac{1}{C_j H_j W_j}\left\|G_j^{\phi}(\hat{\mathbf{y}}) - G_j^{\phi}(\mathbf{y})\right\|_F^2, \qquad G_j^{\phi}(\mathbf{x})_{c,c'} = \frac{1}{C_j H_j W_j}\sum_{h=1}^{H_j}\sum_{w=1}^{W_j}\phi_j(\mathbf{x})_{h,w,c}\phi_j(\mathbf{x})_{h,w,c'},$$

Здесь $\phi_j(\mathbf{x})$ – активация j-того слоя loss network для входа $\mathbf{x}$, $\phi_j(\mathbf{x}) \in \mathbb{R}^{C_j \times H_j \times W_j}$, $G_j^{\phi}(\mathbf{x}) \in \mathcal{M}at(C_j \times C_j)$ – матрица *Грама*.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Обучение Transformation Network

- *Transformation network* обучается с использованием градиентного спуска посредством *минимизации* взвешенной комбинации *функций ошибок*:

$$W^* = \arg\min_W \mathbb{E}_{\mathbf{x},\{\mathbf{y}_i\}} \left[\sum_{i=1}^{M} \lambda_i \ell_i(f_W(\mathbf{x}), \mathbf{y}_i) \right].$$

- *Loss network* при этом *не обучается* и работает как *преобразователь признаков*. Она является *предобученной сетью* на задаче *классификации* изображений (обычно, на *ImageNet*).
- При решении задачи *переноса стиля* $\mathbf{y}_c$ – это *входное* изображение $\mathbf{x}$, а *выход* $\hat{\mathbf{y}}$ должен сочетать в себе $\mathbf{x} = \mathbf{y}_c$ и стиль $\mathbf{y}_s$.
- При решении задачи *увеличения разрешения* изображение $\mathbf{x}$ имеет *низкое* разрешение, а $\mathbf{y}_c$ – *высокое*. *Style reconstruction loss* не используется.

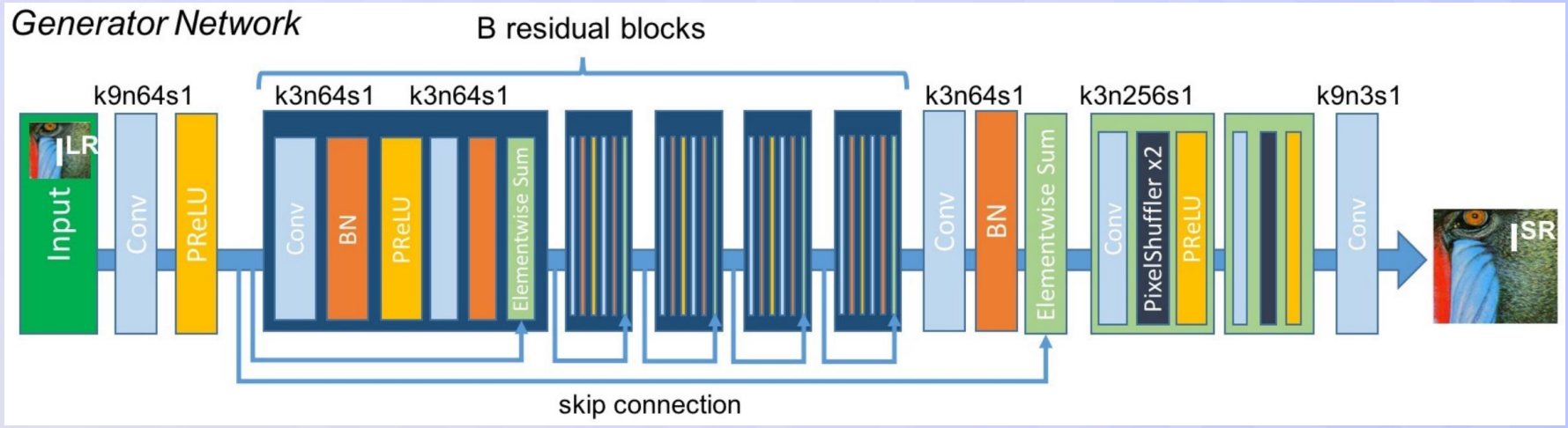
- Аналогичная *идея* положена в основу *Super-Resolution GAN* (*SRGAN*): в процессе генерации изображений с более *высоким разрешением* оценивать их *схожесть* с помощью *perceptual loss*, которая, в свою очередь, вычисляется с использованием *VGG loss* – *евклидова расстояния* между представлениями признаков *сгенерированного* и *реального* изображений, полученных с одного из слоёв *предобученной нейросети VGG-19* (признаки формируются после *активации* j-того сверточного слоя перед i-тым слоем пулинга):

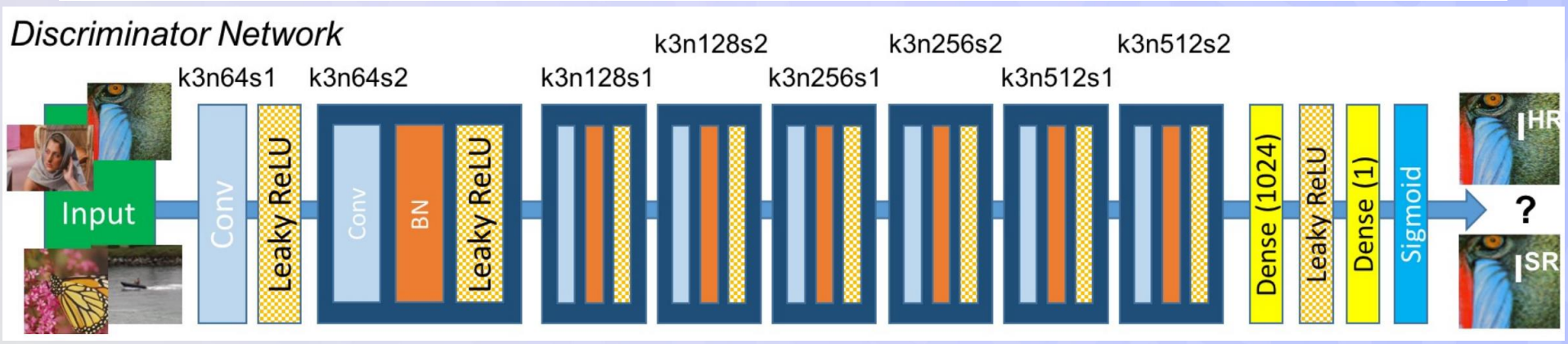
$$\ell^{SR}_{VGG/ij} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} \left[\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}\left(G_{\theta_G}(I^{LR})\right)_{x,y} \right]^2.$$

- Тогда *полная ошибка* будет включать в себя *взвешенную сумму adversarial loss* и *perceptual loss*.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

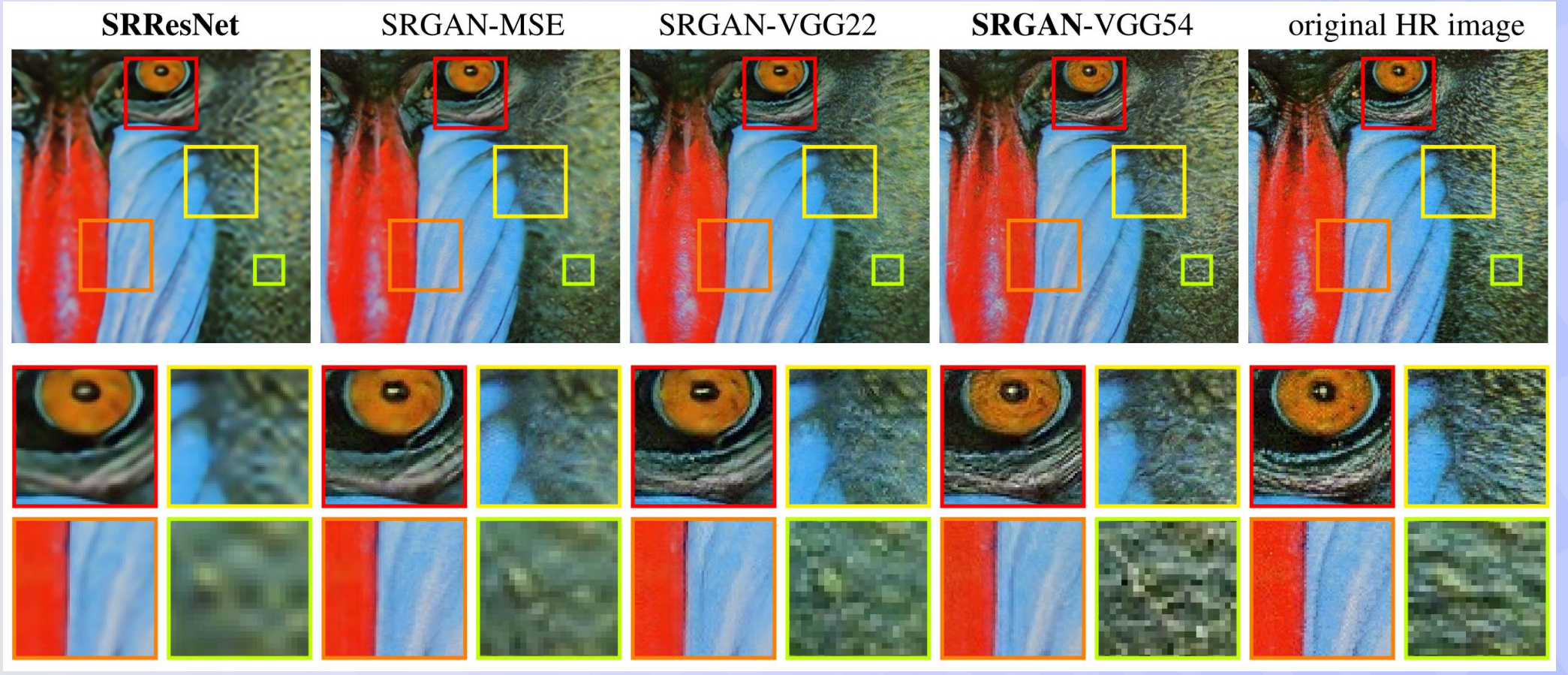
Архитектура генератора и дискриминатора в SRGAN





VOLGOGRAD STATE TECHNICAL UNIVERSITY

Результаты работы SRGAN



VOLGOGRAD STATE TECHNICAL UNIVERSITY

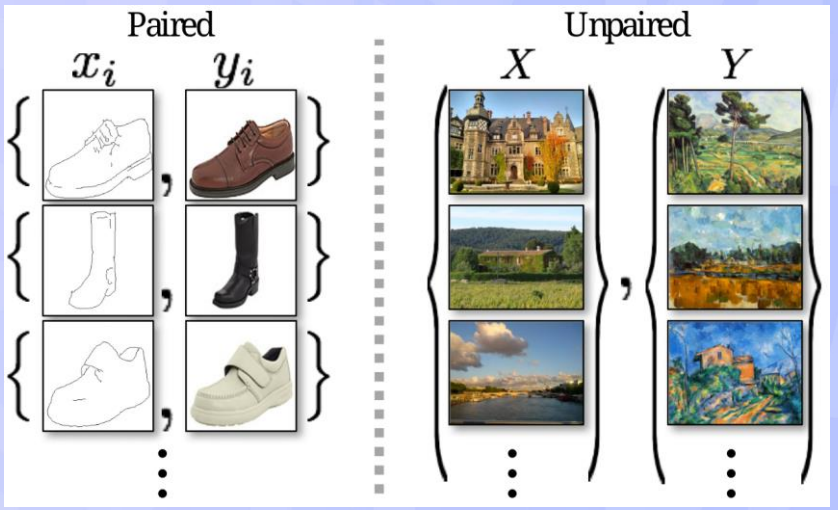
Задачи Image-to-Image Translation

- Задачи Image-to-Image Translation – это *класс задач компьютерного зрения и графики*, целью которого является изучение *отображения* между *входным и выходным* изображениями с использованием обучающего набора соответствующих *пар изображений*.
- Подобная задача широко изучена для различных областей. Она сводится к задаче *обучения «с учителем»* на подготовленных в *датасете* парах изображений: $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$.
- Однако, *подготовка* датасета с парами изображений часто является *сложной и дорогой*. В ряде случаев даже *неясно*, как *определить* выходное изображение.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Cycle-Consistent Adversarial Networks

- Что делать, если *датасет пар недоступен* или имеет *небольшой объём*?
- Для решения задачи было предложено использовать **Cycle-Consistent Adversarial Networks** (*cycleGAN*).
- В основу создания *алгоритма cycleGAN* положена *идея*, что для изучения перевода одного изображения в другое можно использовать не связанные пары изображений, а просто набор изображений из двух классов. При этом алгоритм пытается изучить *связь между классами* изображений.

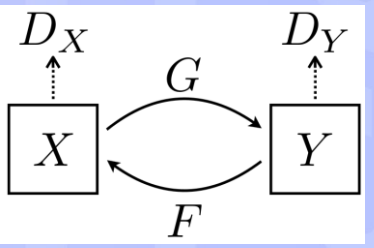


- Обучение *cycleGAN* выполняется на *уровне множеств* X и Y, а не *объектов*.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Свойство постоянства цикла в CycleGAN

- *Изображения*, используемые для *обучения CycleGAN*, принадлежат двум множествам: $\mathbf{x} \in X$, $\mathbf{y} \in Y$.
- Введем отображение $G: X \to Y$, так, чтобы $\hat{\mathbf{y}} = G(\mathbf{x})$ являлся *изображением*, неотличимым от $\mathbf{y}$ с точки зрения *генеративно-состязательного процесса*.
- Тогда $\mathbf{y} \sim p_r(\mathbf{y})$, $\hat{\mathbf{y}} \sim p_r(\mathbf{y})$. Однако, поскольку $\mathbf{x}_i$ и $\mathbf{y}_i$ *не являются парой изображений*, то существует бесконечно много $G : G(\mathbf{x}) \sim p_r(\mathbf{y})$.
- Также, все *недостатки*, свойственные GAN, остаются (например, *mode collapse*). Поэтому, требуется добавить новые ограничения при поиске G.
- В *cycleGAN* вводится свойство постоянства цикла (*cycle consistent*): введём $F: Y \to X$, тогда G и F будут *взаимно обратными*, *биективными* функциями.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

Виды ошибок при обучении CycleGAN

- *Cycle consistent* достигается *одновременным обучением* функций G и F и введением *ошибки постоянства цикла* (*cycle consistency loss*), которая *поощряет* условия $F(G(\mathbf{x})) \approx \mathbf{x}$ и $G(F(\mathbf{y})) \approx \mathbf{y}$:

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{\mathbf{x} \sim p_{rx}(\mathbf{x})}\left[\left\|F(G(\mathbf{x})) - \mathbf{x}\right\|_1\right] + \mathbb{E}_{\mathbf{y} \sim p_{ry}(\mathbf{y})}\left[\left\|G(F(\mathbf{y})) - \mathbf{y}\right\|_1\right].$$

- Для *обучения* CycleGAN также применяется *два вида состязательной ошибки* (*Adversarial Loss*):

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{\mathbf{y} \sim p_{ry}(\mathbf{y})}[\log D_Y(\mathbf{y})] + \mathbb{E}_{\mathbf{x} \sim p_{rx}(\mathbf{x})}[\log(1 - D_Y(G(\mathbf{x})))].$$

$$\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) = \mathbb{E}_{\mathbf{x} \sim p_{rx}(\mathbf{x})}[\log D_X(\mathbf{x})] + \mathbb{E}_{\mathbf{y} \sim p_{ry}(\mathbf{y})}[\log(1 - D_X(F(\mathbf{y})))].$$

- Тогда *полная ошибка* CycleGAN может быть записана в виде:

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

VOLGOGRAD STATE TECHNICAL UNIVERSITY

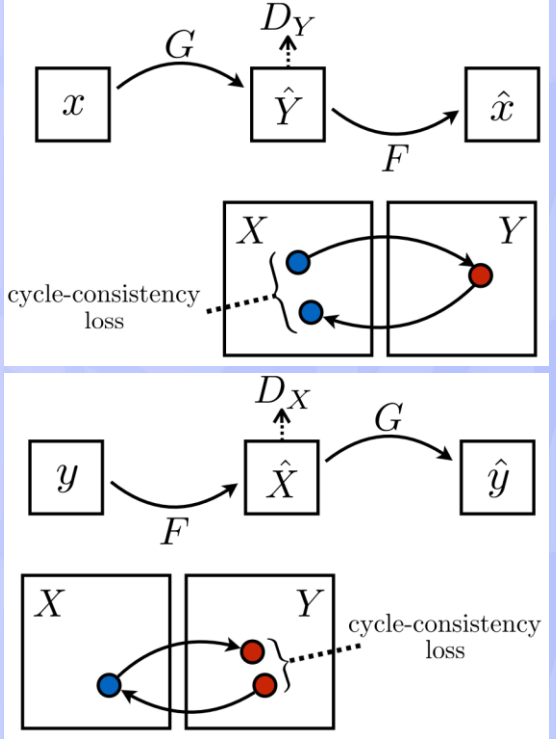
Обучение CycleGAN

- В процессе *обучения* CycleGAN требуется *решить задачу*:

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

- *Процесс обучения* CycleGAN можно представить так:

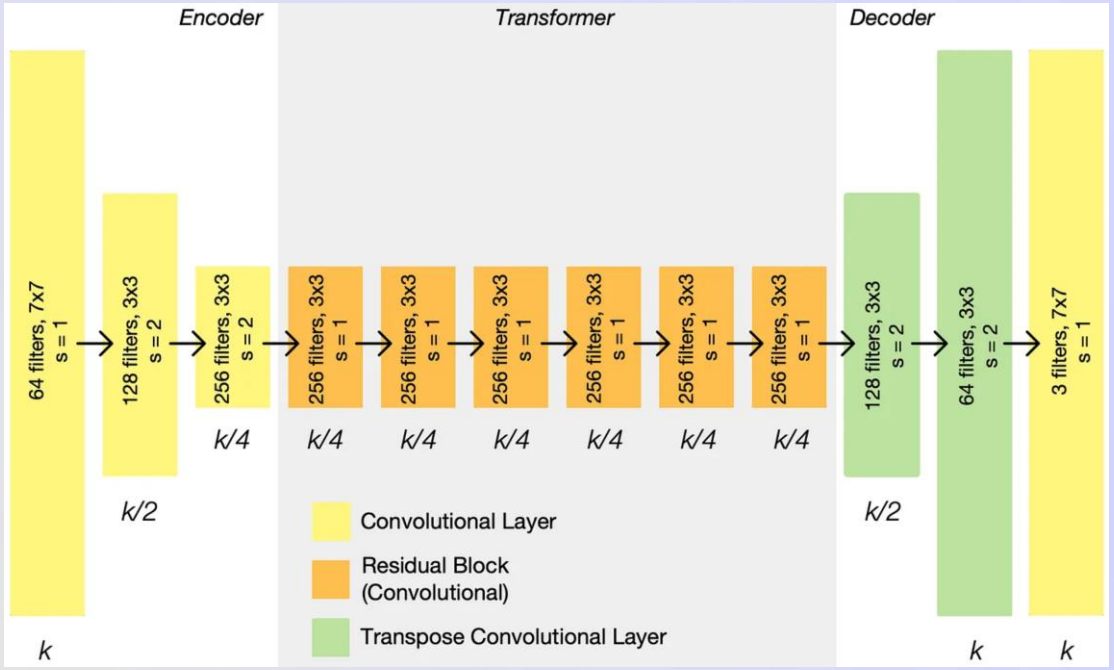
```
for i in iterations:
    # train discriminators
    calc adversarial losses for D_X and D_Y;
    backward() and optimize(D_X, D_Y);
    # train generators
    calc adversarial losses for G and F;
    calc cycle consistent loss, append to losses;
    backward() and optimize(G, F);
```

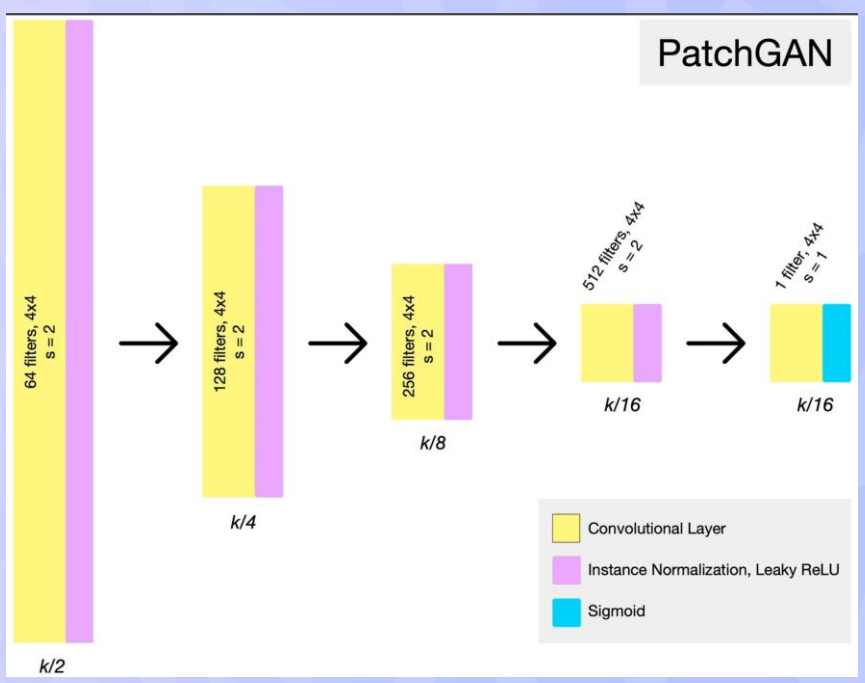


VOLGOGRAD STATE TECHNICAL UNIVERSITY

Классическая реализация CycleGAN

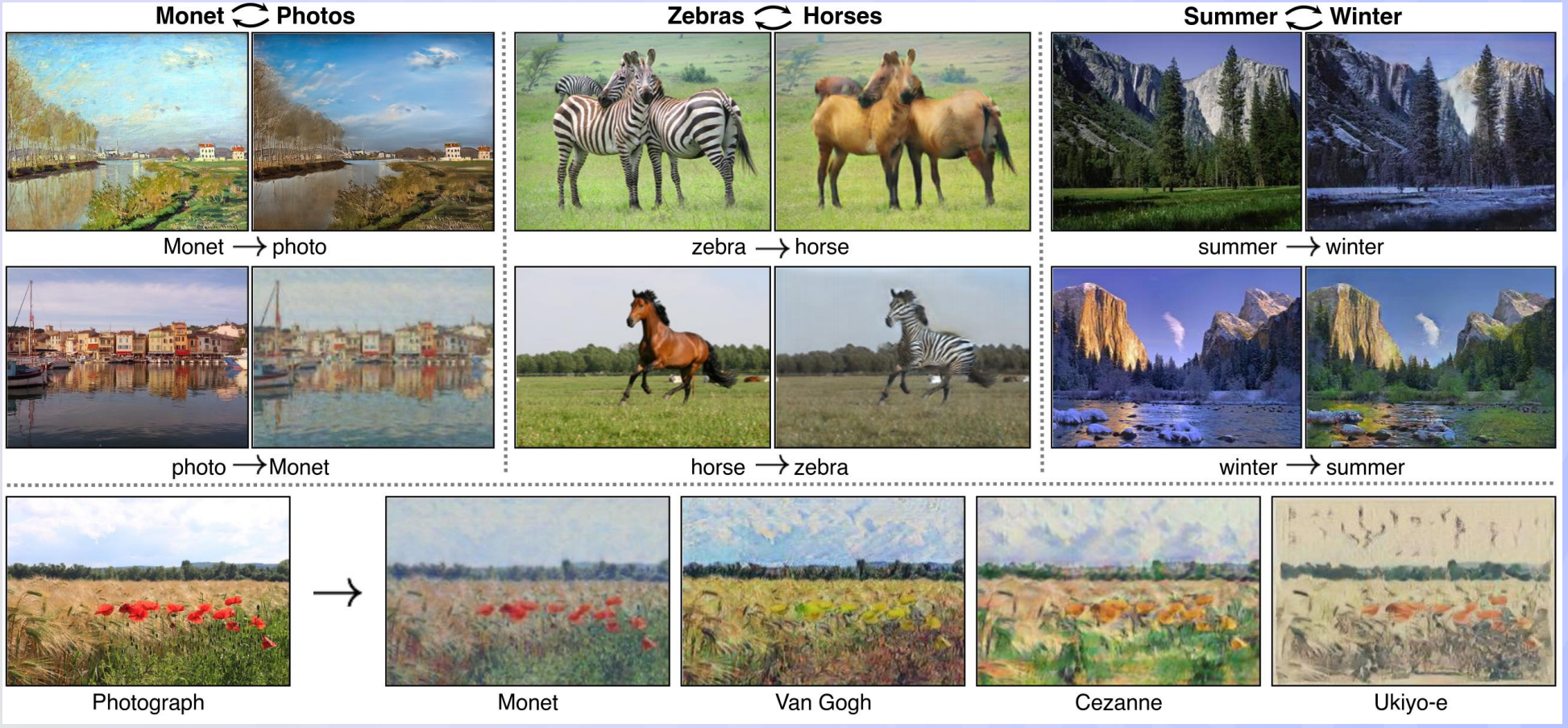
- Использование *MSE* вместо *BCE* показало *лучшие результаты*.
- Обновление дискриминаторов выполняется по истории 50 изображений.





VOLGOGRAD STATE TECHNICAL UNIVERSITY

Результаты работы CycleGAN



VOLGOGRAD STATE TECHNICAL UNIVERSITY

- *Классические* генеративно-состязательные сети ведут себя как «*черные ящики*»: неясно, *как генерируется* изображение, обладающее теми или иными свойствами.
- Свойства *латентного пространства* изучены недостаточно.
- *StyleGAN* предлагает *новую архитектуру генератора* для GAN: *Style-based generator*. В её основе лежит *постепенное* преобразование вектора признаков при *увеличении масштабов* изображения с *добавлением шума*.
- Этот генератор позволяет выполнять *автоматическое*, *неуправляемое* разделение высокоуровневых признаков в генерируемых изображениях.
- *Дискриминатор* и *функция ошибки* остаются неизменными, которые используются в *классических* архитектурах GAN.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Основные операции в StyleGAN

- Введём *обучаемый* латентный вектор $\mathbf{z} \in \mathcal{Z}$ и отображение $f: \mathcal{Z} \to \mathcal{W}$, задаваемое *нейронной сетью* (обычно многослойным перцептроном) (*mapping network*) и формирующее вектор *промежуточного* латентного пространства $\mathbf{w} = f(\mathbf{z}), \mathbf{w} \in \mathcal{W}$.
- Вектор $\mathbf{w}$ посредством *обучаемого аффинного преобразования* преобразуется в *вектор стиля* (*style vector*): $\mathbf{y} = (\mathbf{y}_s, \mathbf{y}_b) = d(\mathbf{w})$, который управляет *адаптивной instance-нормализацией*:

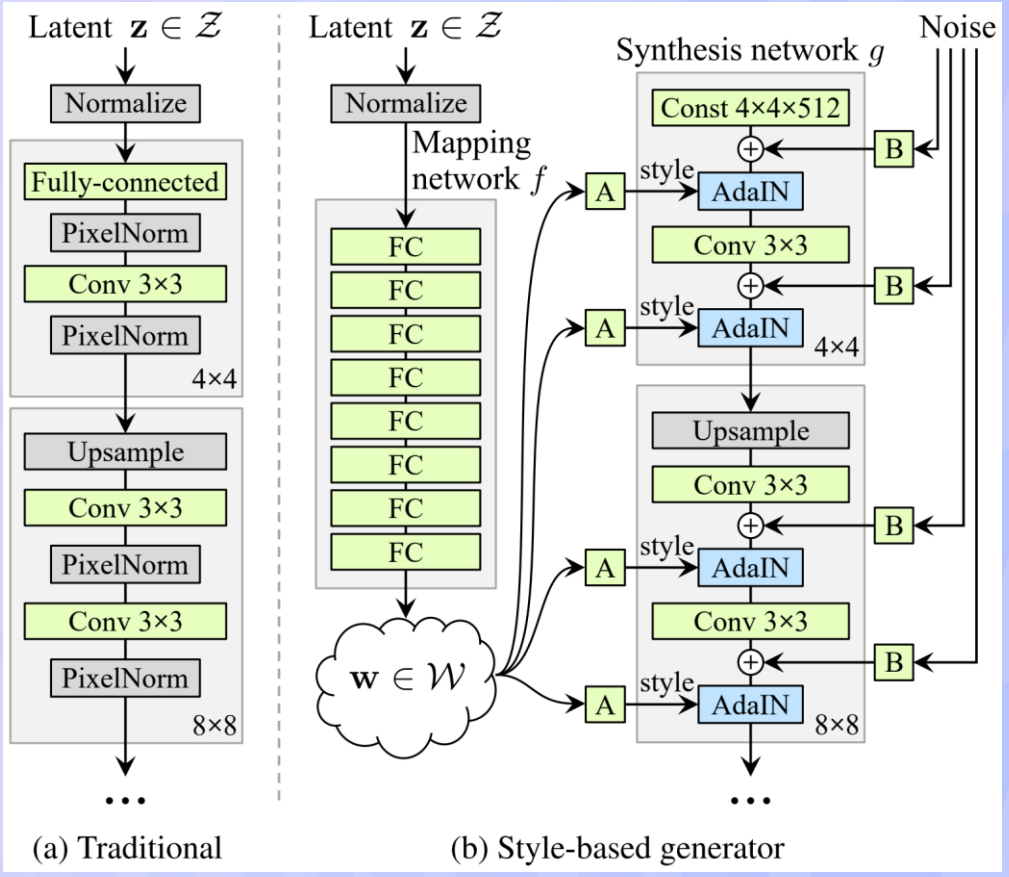
$$\mathrm{AdaIN}(\mathbf{x}_i, \mathbf{y}) = \mathbf{y}_{s,i} \frac{\mathbf{x}_i - \mu(\mathbf{x}_i)}{\sigma(\mathbf{x}_i)} + \mathbf{y}_{b,i}.$$

- На вход *адаптивной instance-нормализации* подаётся также *шум*, который *суммируется* с выходом *предыдущего свёрточного слоя* генератора.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Архитектура StyleGAN

- *Гауссовский шум* добавляется после каждой *свертки*, перед *функцией активации*.
- Здесь "A" обозначает обучаемое аффинное преобразование, а функция "B" применяет обученные коэффициенты масштабирования по каналам к входному шуму.
- *Mapping network f* состоит из 8 слоев, а *synthesis network g* – из 18 слоев, по два слоя на каждое *разрешение*.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

Свойства style-based генератора

- *Архитектура style-based генератора* позволяет управлять синтезом изображений с помощью *модификаций стилей*.
- Можно рассматривать *mapping network* и *аффинные преобразования* как способ получения *образцов для каждого стиля* из изученного распределения, а *synthesis network* – как способ генерирования нового изображения на основе *набора стилей*.
- Эффекты каждого стиля *локализованы* в нейросети, т. е. изменение определенного подмножества стилей может повлиять только на *определенные аспекты* изображения.
- Данные свойства генератора позволяют реализовать *смешение стилей* (*style mixing*) и *стохастическую вариацию изображения*.

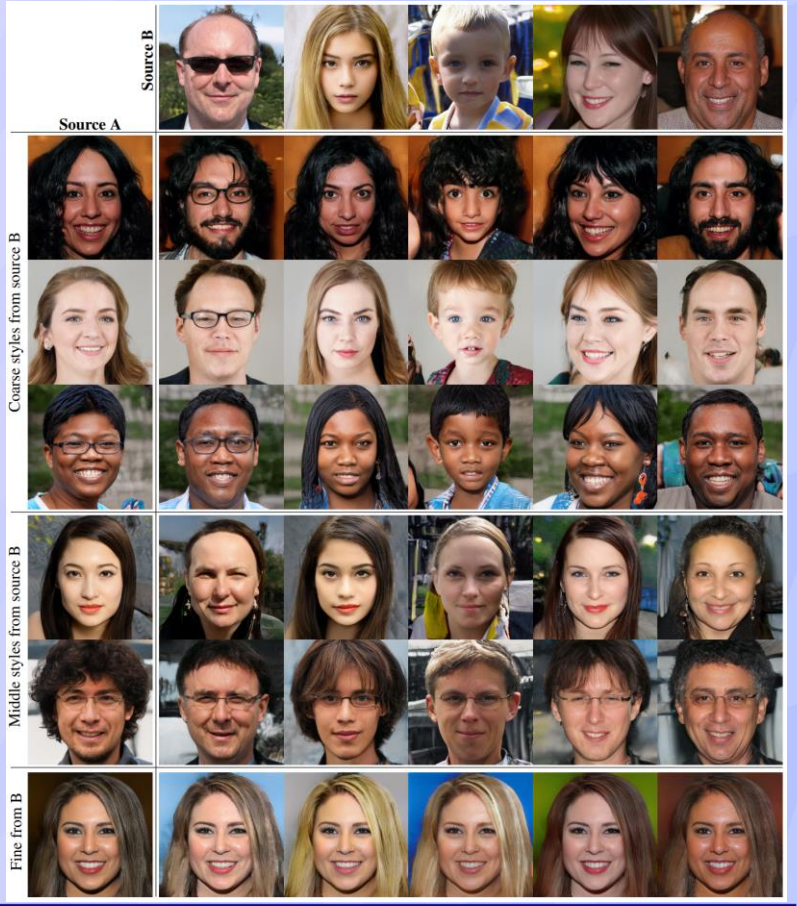
VOLGOGRAD STATE TECHNICAL UNIVERSITY

Смешение стилей в StyleGAN

- Для создания изображения с *различными стилями* нужно просто переключиться с *одного* латентного вектора на *другой* в *произвольно выбранной* точке *synthesis network*.
- $\mathbf{w}_1 = f(\mathbf{z}_1), \mathbf{w}_2 = f(\mathbf{z}_2), \xi_i = \mathrm{AdaIN}\left(\mathbf{x}_i, \mathbf{y}\left(\mathbf{w}_{k(i)}\right)\right),$

$$u = \mathrm{rand}(1, N), \qquad k(i) = \begin{cases} 1, & \text{if } i < u \\ 2, & \text{otherwise} \end{cases}$$

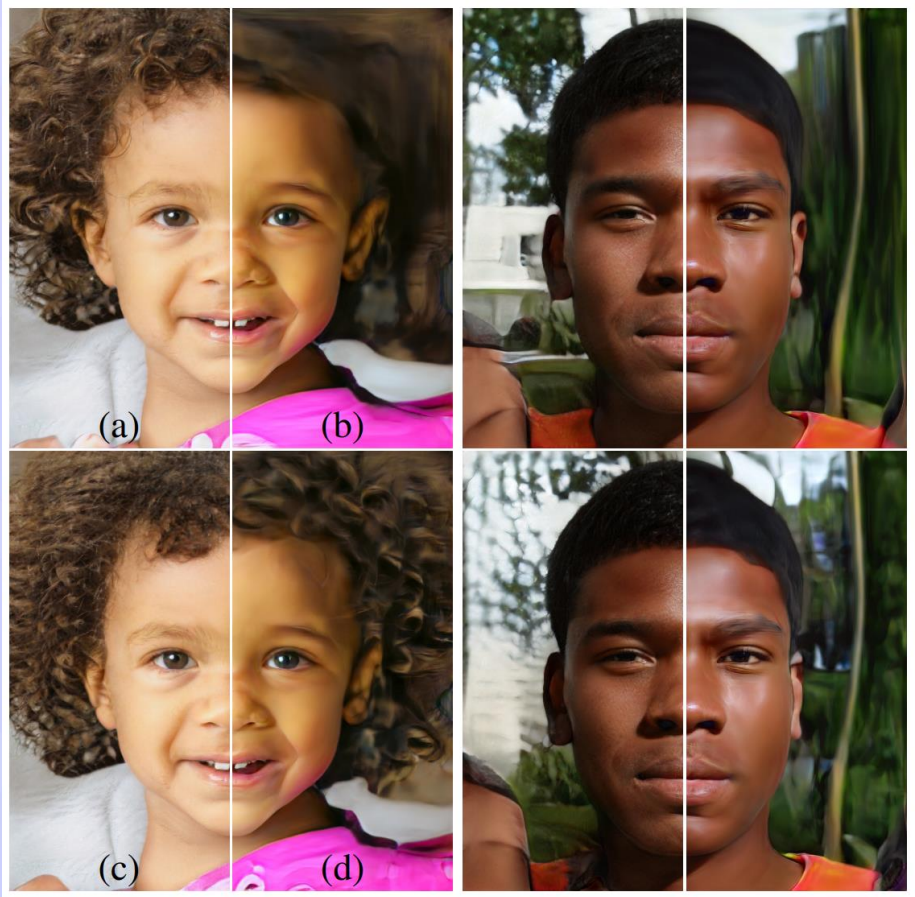
- Чем раньше в *synthesis network* включается латентный вектор $\mathbf{w}_2$, отвечающий за *второй стиль*, тем больше элементов из *второго стиля* появится в *генерируемом* изображении.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

Стохастическая вариация в StyleGAN

- *Эффект шума*, подаваемого на слои сети:
  a) Шум, поданный на *все* слои.
  b) Отсутствие шума.
  c) Шум, поданный на *поздние* слои.
  d) Шум, поданный на *ранние* слои.
- Видно, что отсутствие шума ведет к "*живописному*" виду. Шум на *ранних слоях* приводит к крупным завиткам волос и появлению крупных фоновых деталей, а на *поздних* – выявляет более мелкие завитки волос, мелкие детали фона и поры кожи.



VOLGOGRAD STATE TECHNICAL UNIVERSITY

Демонстрация практических примеров

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Заключение

1. Рассмотрели новую меру схожести распределений и выявили её особенности, значимые для обучения генеративно-состязательных сетей.
2. Поговорили по Wasserstein GAN и рассмотрели алгоритм его обучения.
3. Рассмотрели алгоритм спектральной нормализации.
4. Рассмотрели задачи преобразования изображений, поговорили о подходах к её решению, выяснили место GANs среди них.
5. Поговорили про использование perception loss при работе со стилями и разрешением изображений, рассмотрели архитектуру SRGAN.
6. Рассмотрели архитектуру, функционирование и способ обучения CycleGAN.
7. В теории и на практическом примере рассмотрели принцип обучения StyleGAN и генерацию изображений с её использованием.

VOLGOGRAD STATE TECHNICAL UNIVERSITY

Спасибо за внимание!

Волгоград 2025