

Прикладные задачи анализа данных

Лекция 1
Трансформация изображений

Евгений Соколов
esokolov@hse.ru

НИУ ВШЭ, 2022

О курсе

- Генеративные методы
- Компьютерное зрение
- Глубинное обучение для обработки звука
- Задачи в NLP
- Рекомендательные системы
- Временные ряды
- Продуктовая аналитика
- ML Ops

О курсе

Итог = Округление(0.5 * ДЗ + 0.2 * КР + 0.3 * Э)

Что было раньше

- Классификация изображений
 - Сегментация
 - Детекция
 - Идентификация
-
- Везде выход модели — класс для всей картинки, какой-то её части или для каждого пикселя

Что будет дальше

- Выход модели — изображение
- Основная проблема: выход действительно должен выглядеть как изображение
- Superresolution
- Inpainting
- Перенос стиля
- Генерация изображений

Метрики сходства изображений и перенос стиля

Евклидово расстояние



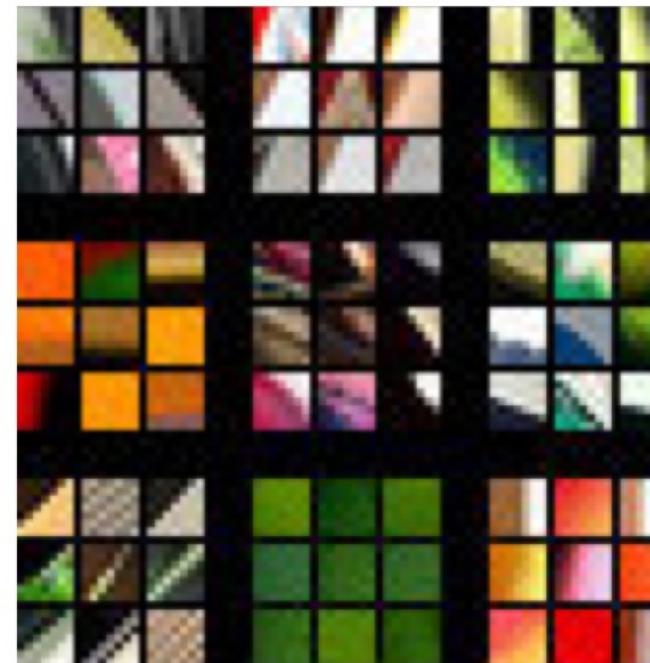
Евклидово расстояние

- При небольших сдвигах или возмущениях может радикально измениться
- Нужен другой способ сравнения изображений

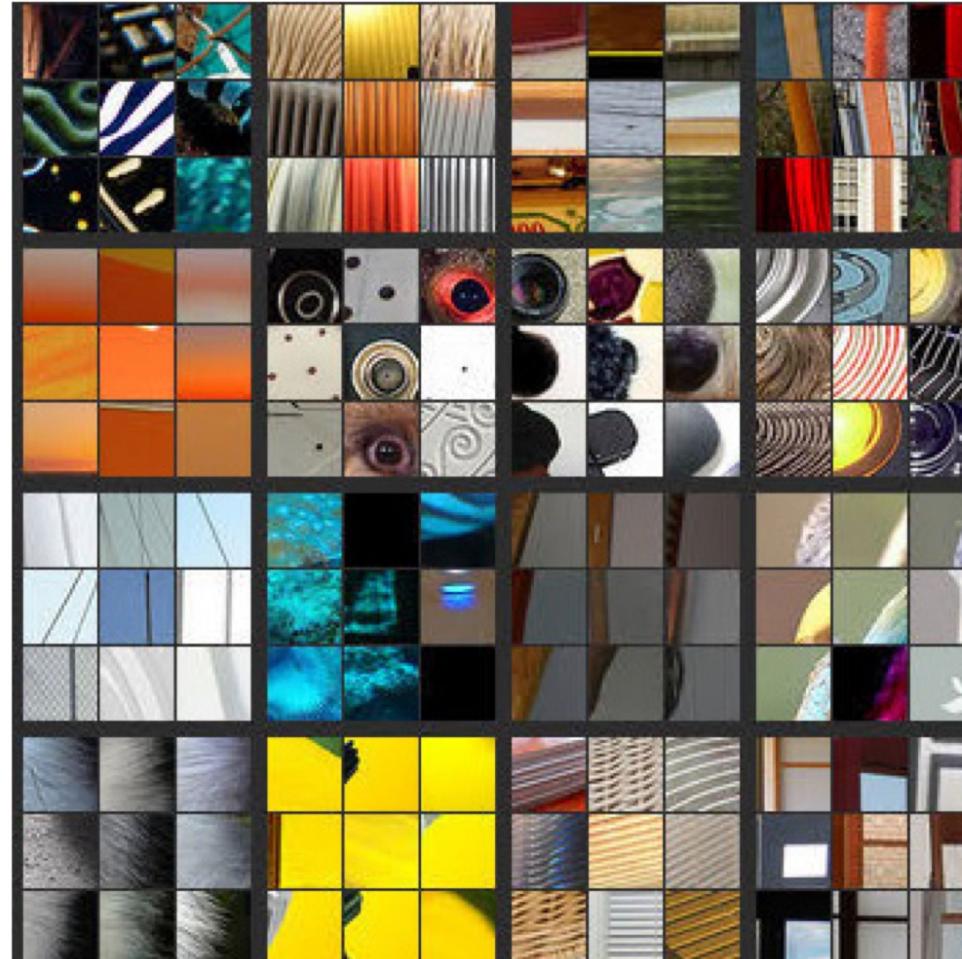
VGG (2014)

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

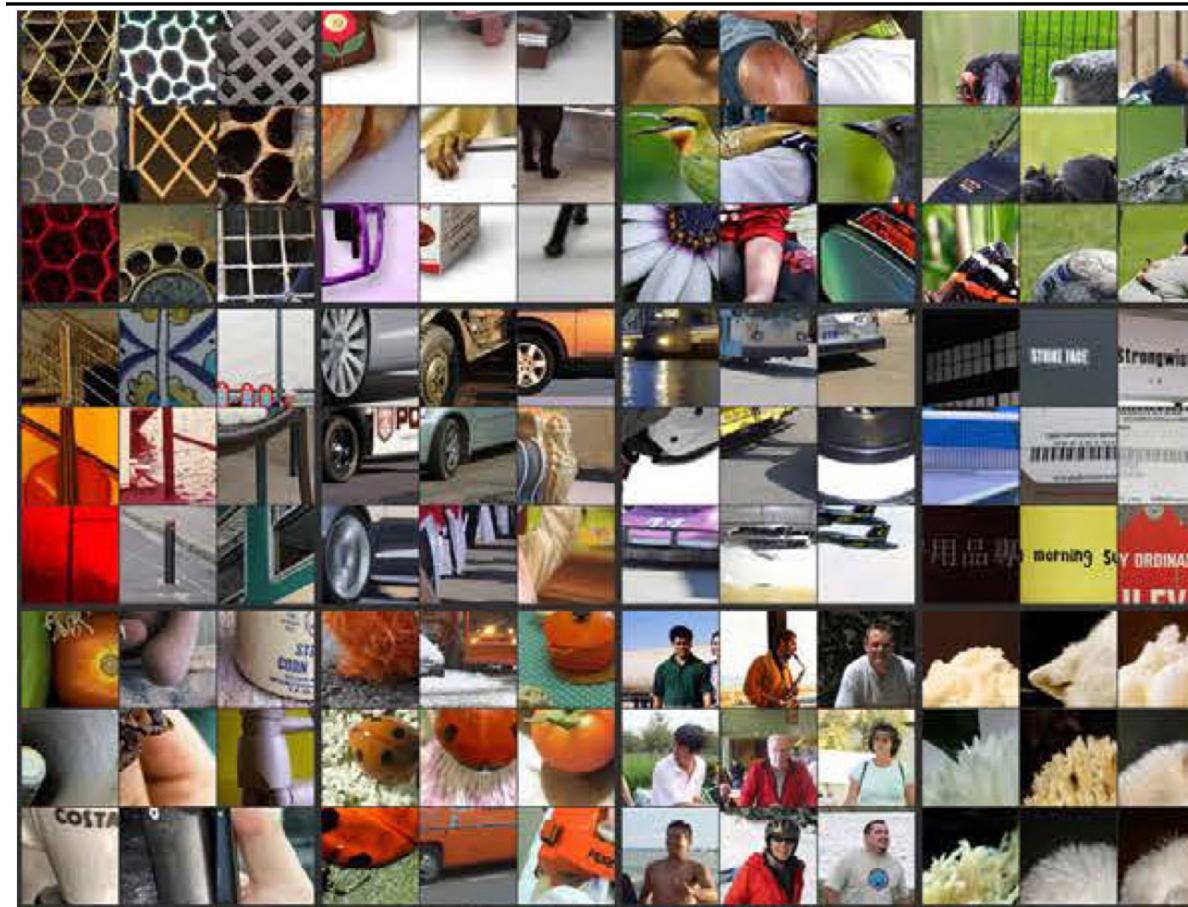
Слой 1



Слой 2



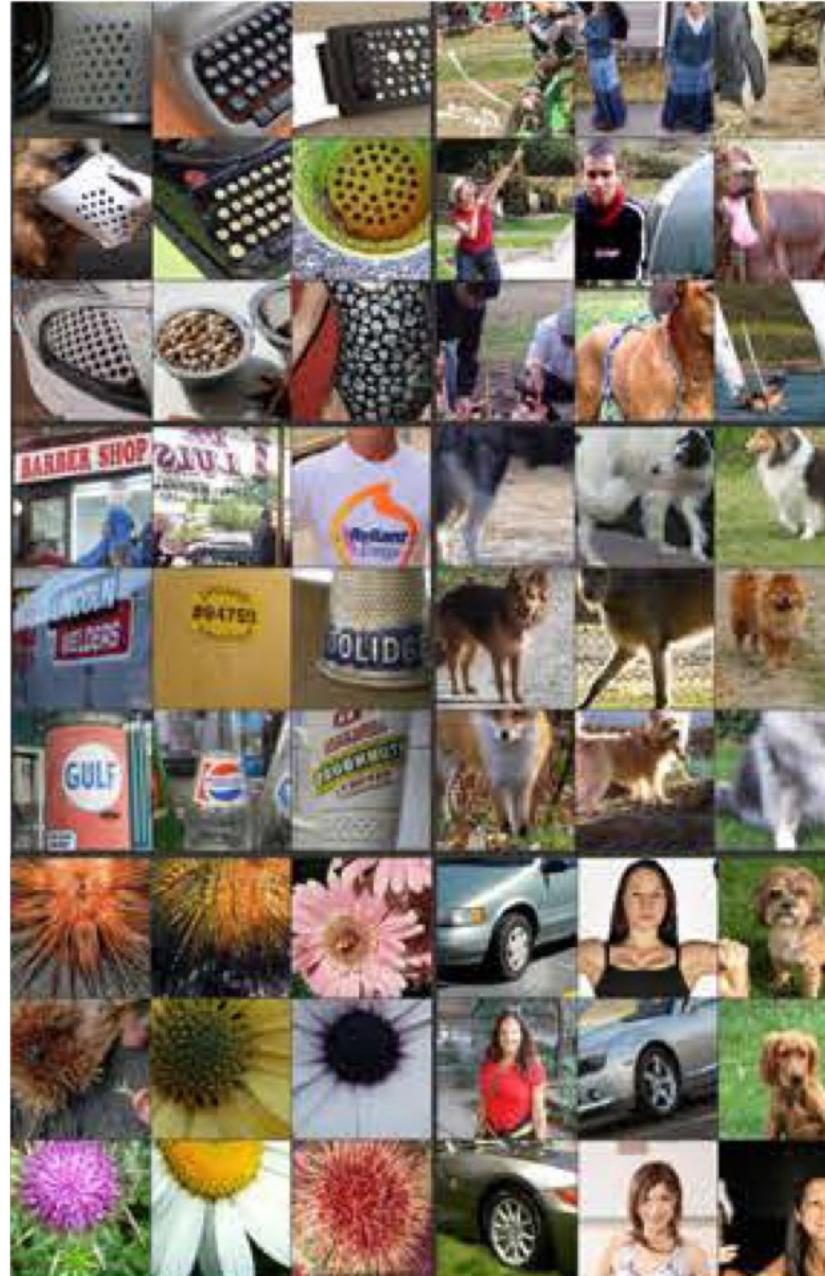
Слой 3



Слой 4



Слой 5



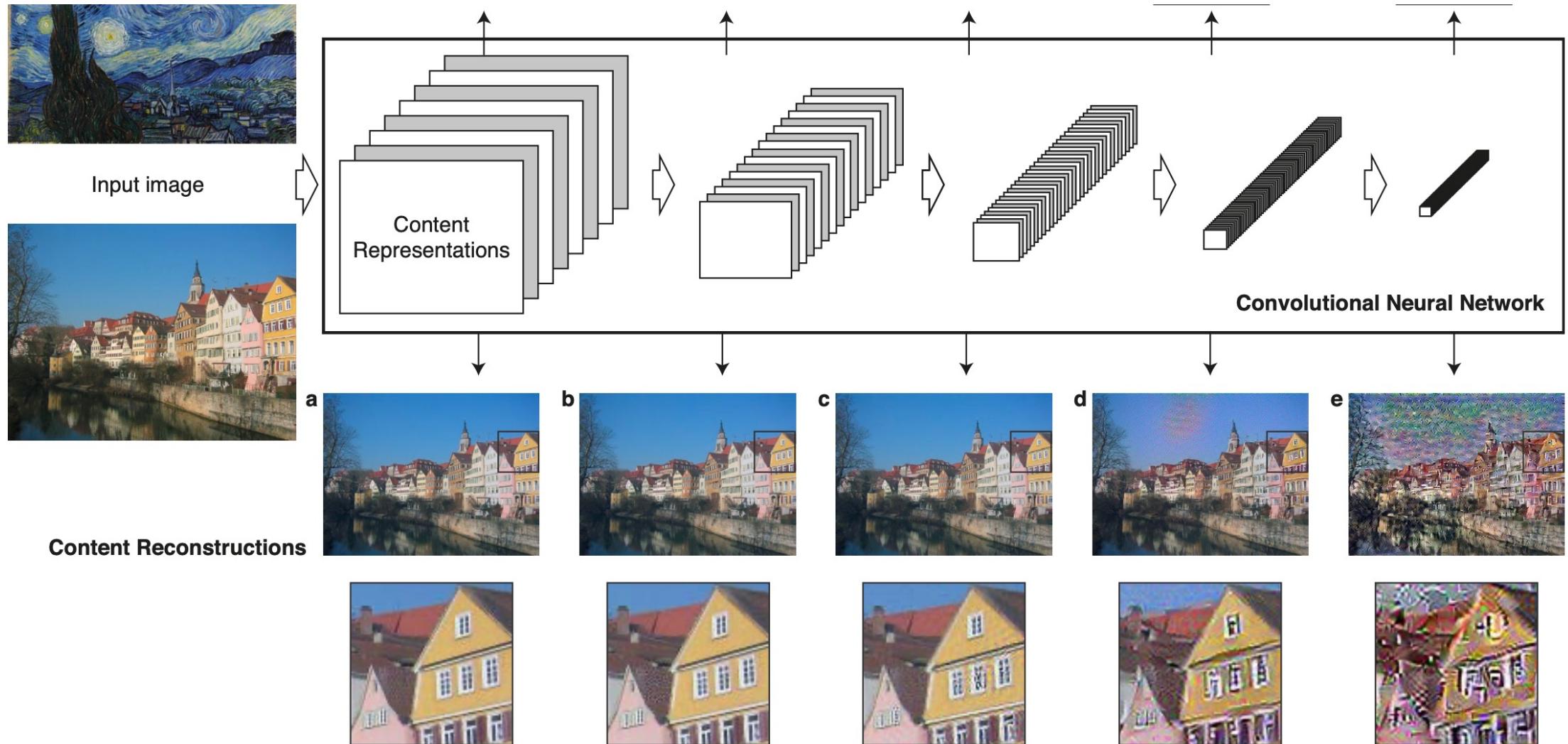
Perceptual loss

- Разница в контенте изображений:

$$L_{\text{content}}^{\ell}(A, B) = \sum_{i,j} (A_{ij}^{\ell} - B_{ij}^{\ell})^2$$

- A_{ij}^{ℓ} — значение i -го фильтра на j -й позиции в слое ℓ для изображения A
- Чем дальше слой, тем меньше он чувствителен к небольшим изменениям в изображении
- Можем попробовать градиентным спуском из картинки с белым шумом найти изображение, минимизирующее $L_{\text{content}}^{\ell}(A, B)$

Perceptual loss



Perceptual loss

- Разница в стиле:

$$L_{\text{style}}^{\ell}(A, B) = \sum_{i,j} \left(G_{ij}^{\ell}(A) - G_{ij}^{\ell}(B) \right)^2$$

- $G_{ij}^{\ell}(A) = \sum_k A_{ik}^{\ell} A_{jk}^{\ell}$ — скалярное произведение i -го и j -го каналов в слое ℓ для изображения A

$$L_{\text{style}}(A, B) = \sum_{\ell=0}^L w_{\ell} L_{\text{style}}^{\ell}(A, B)$$

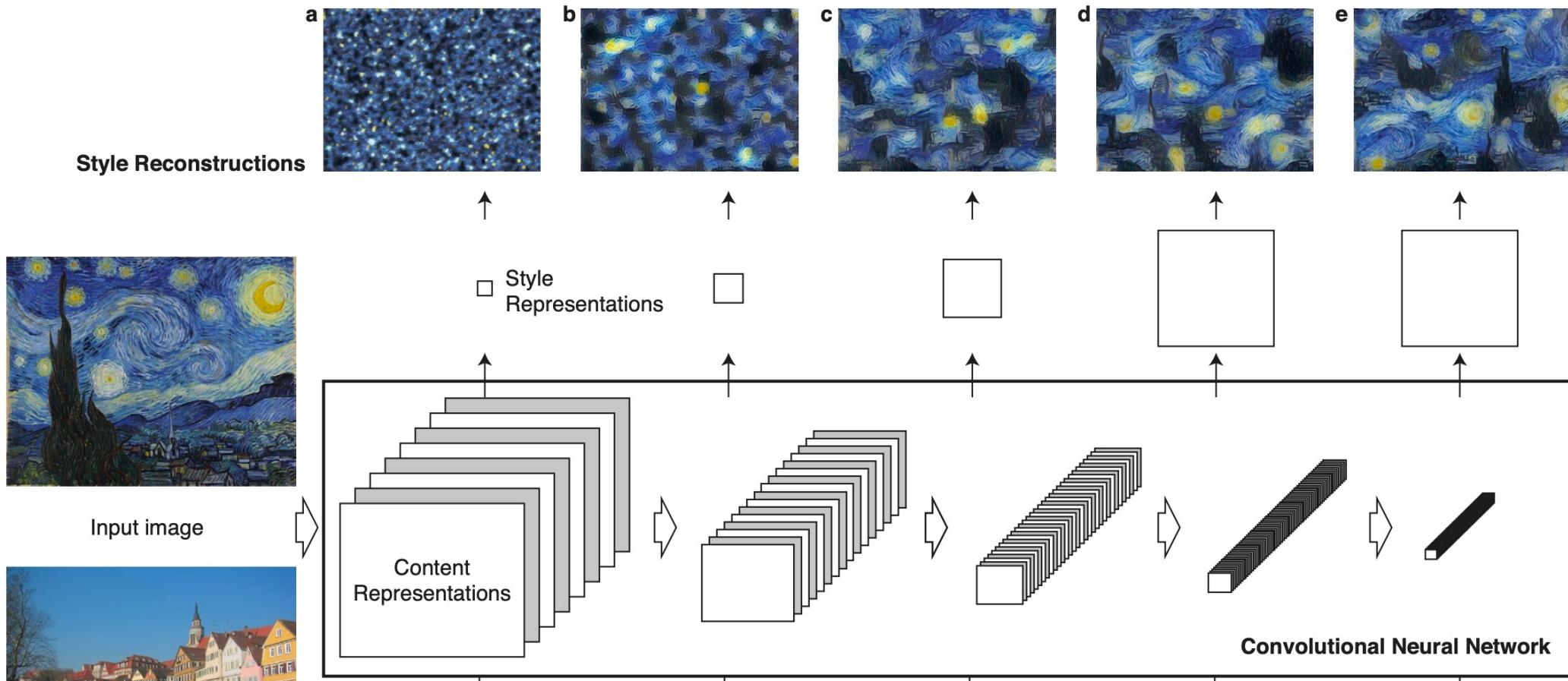
Perceptual loss

- $G_{ij}^\ell(A) = \sum_k A_{ik}^\ell A_{jk}^\ell$ — скалярное произведение i -го и j -го каналов в слое ℓ для изображения A

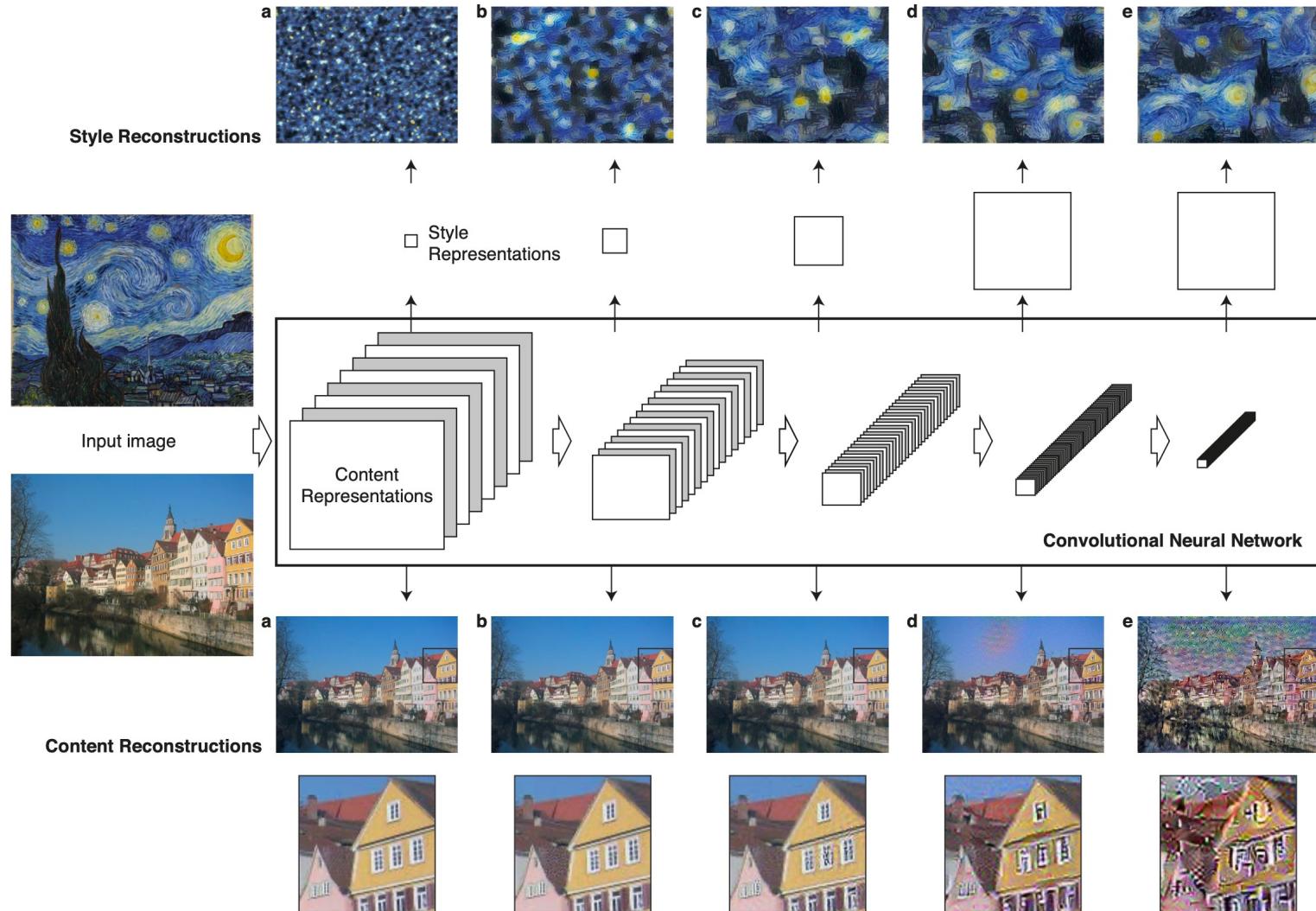
Пример:

- Фильтр 1 детектирует человека
- Фильтр 2 детектирует что-то синее
- Если стилевая картинка синяя, то G_{12} будет высоким
- При переносе стиля будет пытаться сделать всё синим, чтобы увеличить G_{12}

Perceptual loss



Perceptual loss



Perceptual loss

$$L(C, S, X) = \alpha L_{\text{content}}(C, X) + \beta L_{\text{style}}(S, X)$$

- C — исходное изображение
- S — стилевое изображение
- X — итоговое изображение (должно по содержанию быть похожим на C , а по стилю на S)

Perceptual loss

$$L(C, S, X) = \alpha L_{\text{content}}(C, X) + \beta L_{\text{style}}(S, X)$$

- Функции потерь вычисляются с помощью предобученной сети VGG-16
- Для контентной части используется последний свёрточный слой
- Для стилевой части используются слои с 1 по 5 с равными весами

$$\frac{\alpha}{\beta} = 8 * 10^{-4}$$



$$\frac{\alpha}{\beta} = 1 * 10^{-3}$$



$$\frac{\alpha}{\beta} = 5 * 10^{-3}$$

$$\frac{\alpha}{\beta} = 5 * 10^{-4}$$

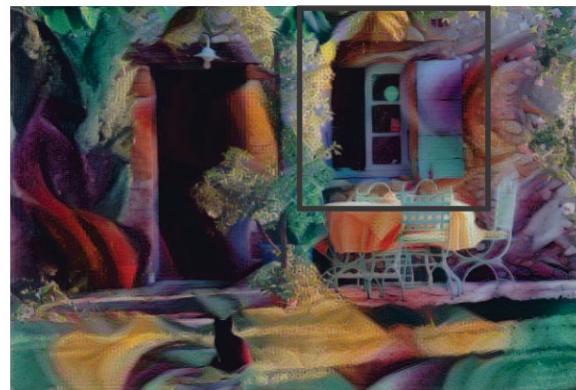


$$\frac{\alpha}{\beta} = 5 * 10^{-4}$$

Content Image



Conv2_2



Conv4_2



Проблемы

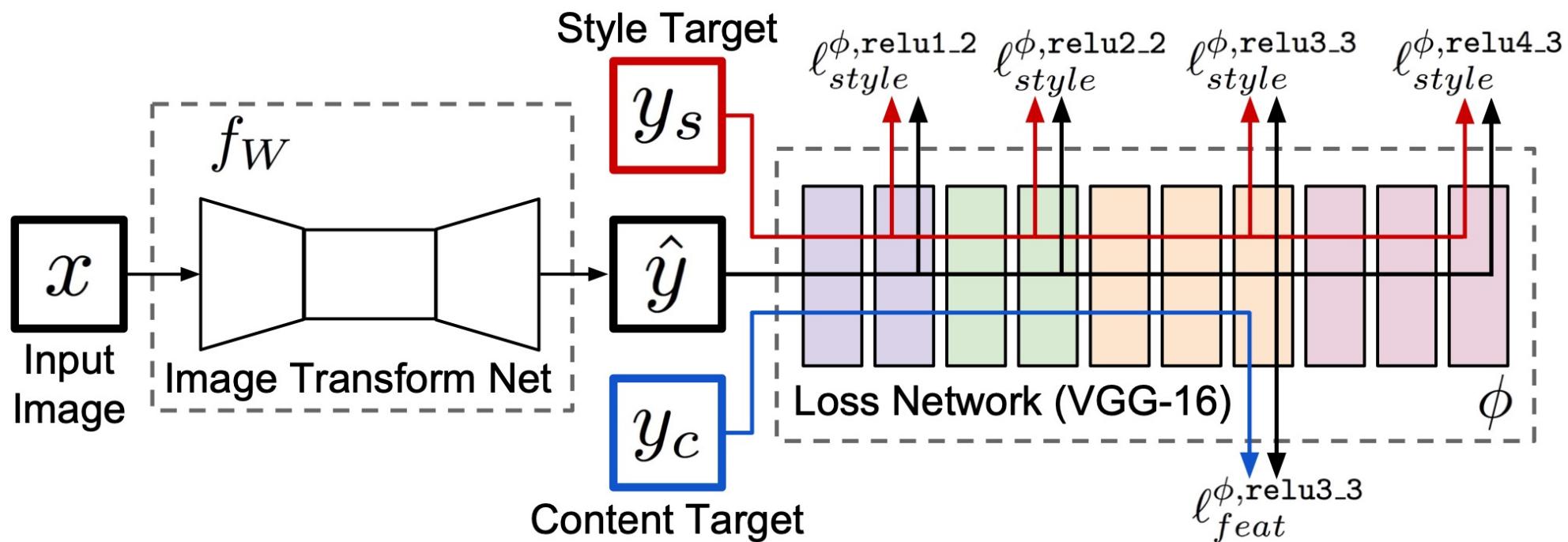
$$L(C, S, X) = \alpha L_{\text{content}}(C, X) + \beta L_{\text{style}}(S, X)$$

- Чтобы найти X , надо посчитать градиент функции потерь по нему много раз
- Вычисление градиента включает в себя обратное распространение ошибки по огромной сети
- 15 секунд для изображения 256x256
- 214 секунд для изображения 1024x1024

Ускорение переноса стиля

- Зафиксируем стилевое изображение S
- Обучим модель $a(C)$ на выборке так, чтобы её выход минимизировал perceptual loss

Ускорение переноса стиля



Ускорение переноса стиля

Image Size	Gatys <i>et al</i> [10]			Ours	Speedup		
	100	300	500		100	300	500
256 × 256	3.17	9.52s	15.86s	0.015s	212x	636x	1060x
512 × 512	10.97	32.91s	54.85s	0.05s	205x	615x	1026x
1024 × 1024	42.89	128.66s	214.44s	0.21s	208x	625x	1042x

Ускорение переноса стиля

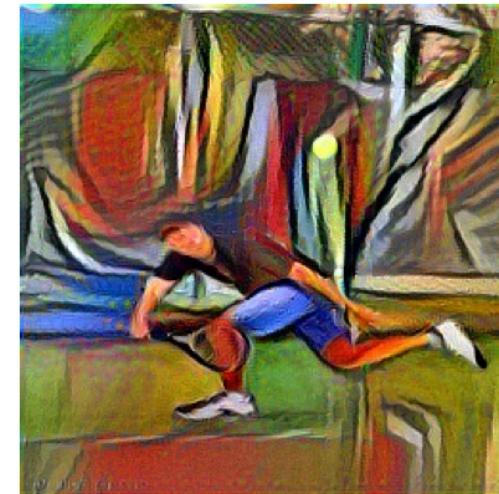
Style



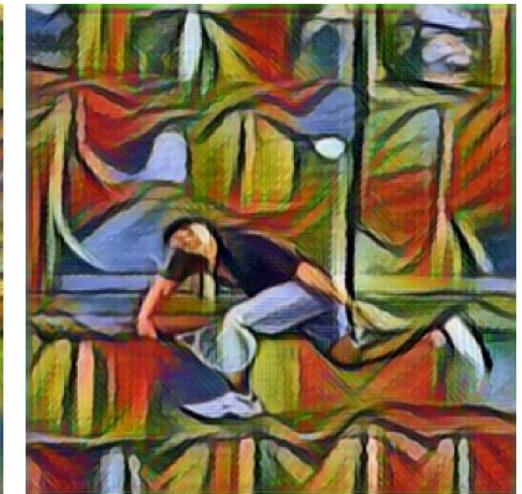
Content



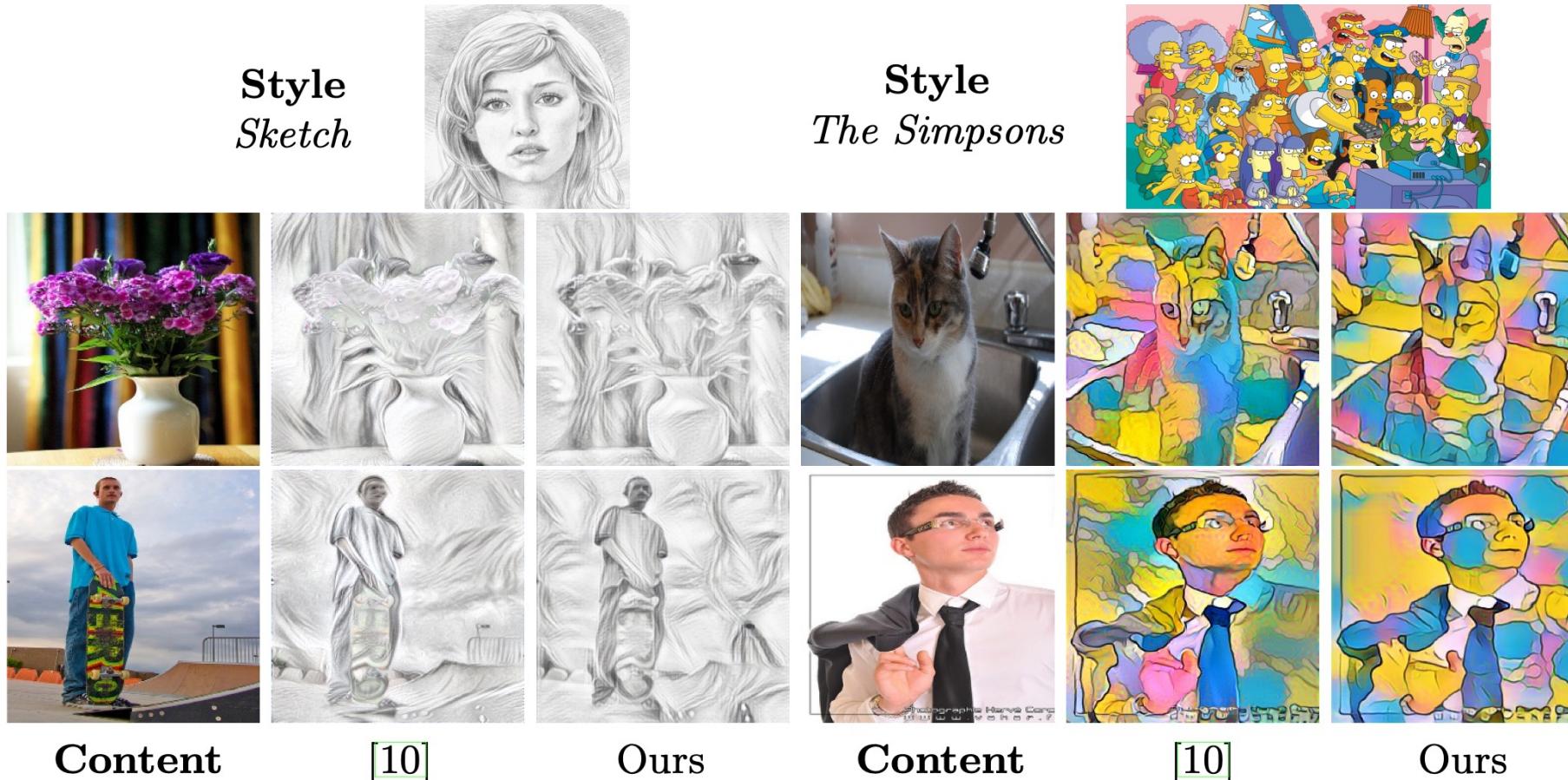
Gatys *et al* [10]



Ours



Ускорение переноса стиля



Superresolution

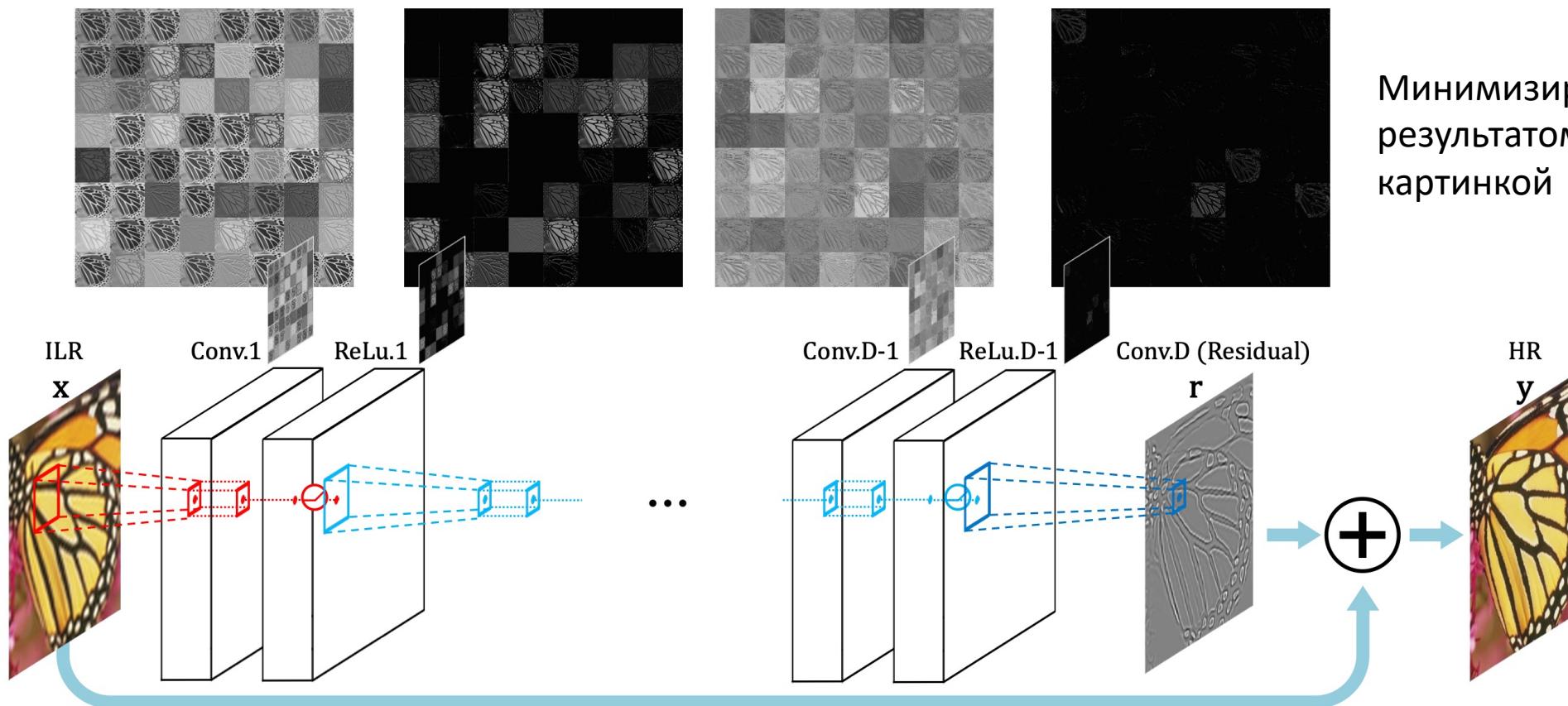
Повышение разрешения



Повышение разрешения

- На входе и на выходе изображения
- Можно генерировать обучающие выборки неограниченного размера

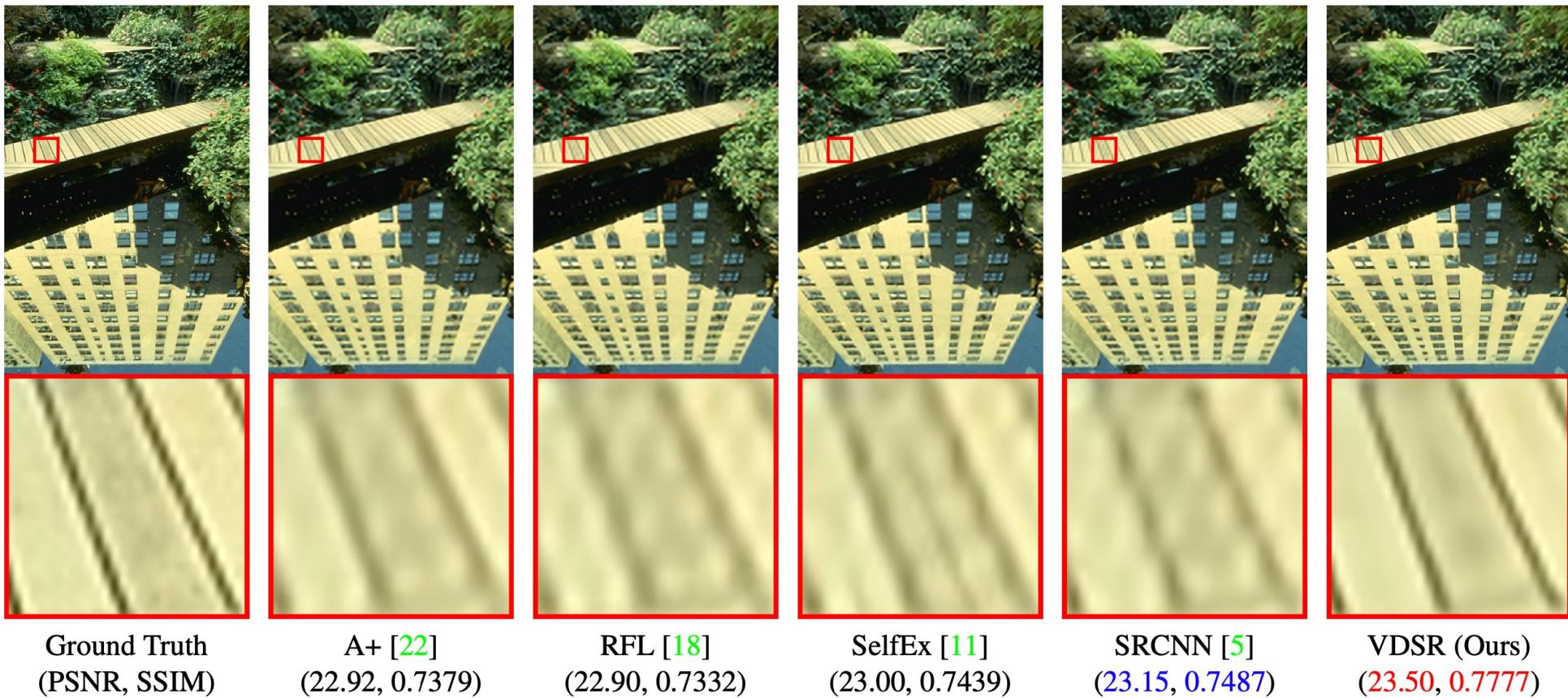
Подход в лоб



Подход в лоб

4.1. The Deeper, the Better

Подход в лоб



Подход с perceptual loss



	Ground Truth	Bicubic	Ours (ℓ_{pixel})	SRCNN [11]	Ours (ℓ_{feat})
This image	31.78 / 0.8577	31.47 / 0.8573	32.99 / 0.8784	29.24 / 0.7841	
Set5 mean	28.43 / 0.8114	28.40 / 0.8205	30.48 / 0.8628	27.09 / 0.7680	

Подход с perceptual loss



	Ground Truth	Bicubic	Ours (ℓ_{pixel})	SRCCN [11]	Ours (ℓ_{feat})
This Image	21.69 / 0.5840	21.66 / 0.5881	22.53 / 0.6524	21.04 / 0.6116	
Set14 mean	25.99 / 0.7301	25.75 / 0.6994	27.49 / 0.7503	24.99 / 0.6731	
BSD100 mean	25.96 / 0.682	25.91 / 0.6680	26.90 / 0.7101	24.95 / 63.17	

Inpainting

Восстановление изображения



(a)

(b)

(c)

(d)

Восстановление изображения

