

# Style-transfer and texture generation

Николай Долженко

ФКН ПМИ

15 марта 2019

# Оглавление

- Вступление
- Обзор подходов
- Neural style
- Texture networks
- Сравнение моделей

# Вступление

# Texture synthesis

Генерация новых примеров текстуры с опорой на уже имеющуюся текстуру



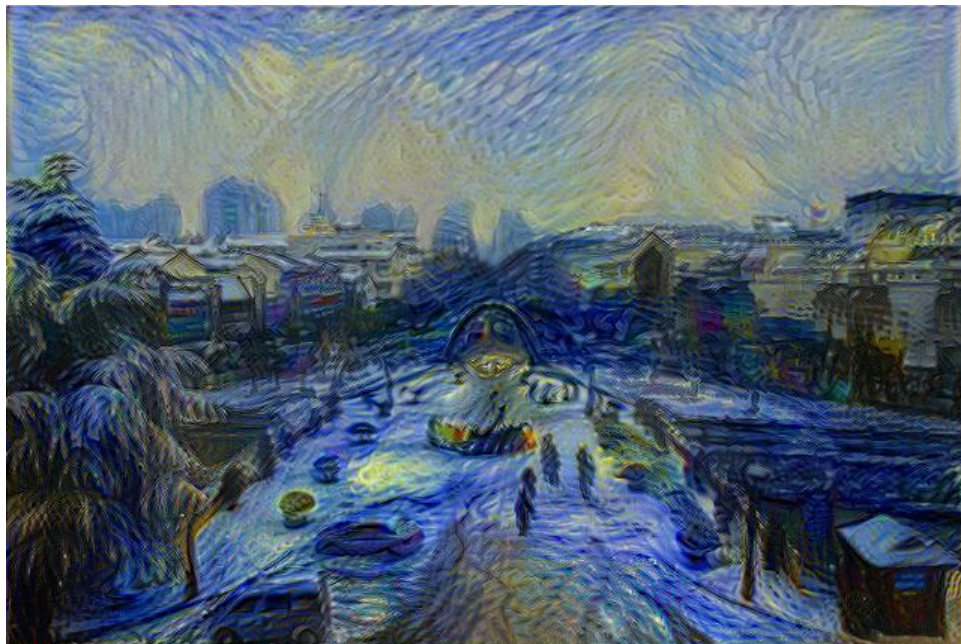
<https://arxiv.org/pdf/1603.03417.pdf>

# Style-transfer

Перенос стиля из одного изображения на другое без значительного изменения содержимого



+



<https://github.com/fzliu/style-transfer>

# Формальная постановка задачи

Задача генерации изображения  $x$  — сэмплирование некоторого распределения  $p(x)$ .

Тогда задача синтеза текстуры  $x_0$  — сэмплирование  $p(x|x_0)$

Задача переноса стиля с изображения  $x_s$  на изображение  $x_o$  — сэмплирование  $p(x|x_o, x_s)$

# Зачем это нужно?

## Style Transfer:

1. Обработка фотографий и видео
2. Аугментация данных

## Texture generation:

1. Компьютерная графика
2. Обработка изображений
3. Пост-продакшн фильмов



<https://arxiv.org/pdf/1809.05375.pdf>

# Обзор подходов



# Descriptive models

Модель учится создавать статистики (описания)

Далее сэмплируются изображения, удовлетворяющие этим статистикам

Примеры: FRAME (Zhu et. al 1998), Neural style (Gatys et. al 2015)

# Generative models

Идея подхода заключается в том, чтобы сразу обучать генерирующий алгоритм  $g(z)$  такой, что  $g(z) \sim X$ .

- $z$  может быть детерминированным набором параметров, тогда нужна обучающая выборка  $(z, x)$
- $z$  может быть шумом, но тогда нужно уметь оценивать качество  $g(z)$

# Generative models

GAN (General Adversarial Networks):

Учит собственную функцию потерь

MMN (Moment matching networks):

Для набора статистик  $\mu_p = E_{\mathbf{x} \sim p(\mathbf{x})}[\phi(\mathbf{x})]$  однозначно задает распределение.

Можем попытаться минимизировать разность между  $\frac{1}{m} \sum_{i=1}^m \phi \circ \mathbf{g}(\mathbf{z}_i)$   
и  $\frac{1}{m} \sum_{i=1}^m \phi(\mathbf{x}_i)$

Neural style

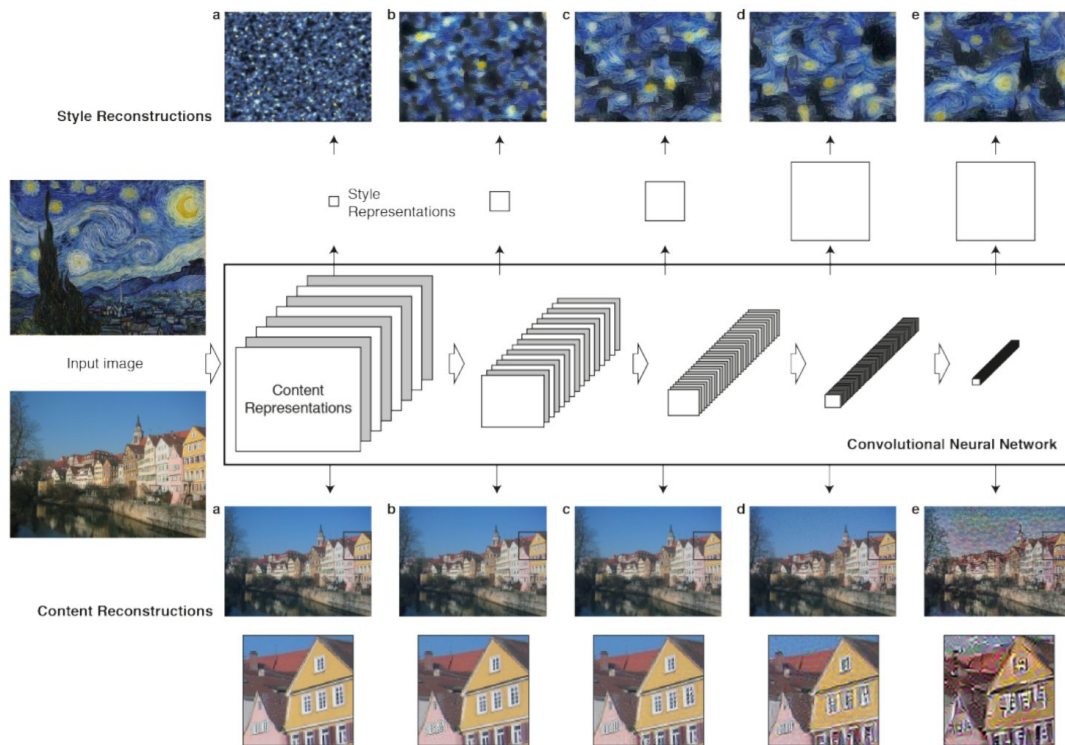
# Идея

Сверточные нейронные сети учат и семантическое содержание, и стиль изображения.

Для задачи синтеза текстур использовать только style loss, для style transfer использовать style loss и content loss

Из такой сети, как VGG, можно получить набор статистик для сравнения сходства изображений.

# Идея



# Идея

Эти две статистики слабо зависят друг от друга

А значит, мы можем искать изображение подходящее под стиль одного и содержание другого

# Реализация

Пусть  $p$  — изображение с содержанием,  $x$  — генерируемое изображение.

$F$  — матрица вытянутых карт признаков (feature map) на слое  $l$  для  $x$

$P$  — матрица вытянутых карт признаков на слое  $l$  для  $p$

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

$$\frac{\partial \mathcal{L}_{content}}{\partial F_{ij}^l} = \begin{cases} (F^l - P^l)_{ij} & \text{if } F_{ij}^l > 0 \\ 0 & \text{if } F_{ij}^l < 0 \end{cases}$$



# Реализация

G — Матрица Грама F

A — Матрица Грама P

N — Количество фильтров в слое l

M — Количество элементов в одной карте признаков

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

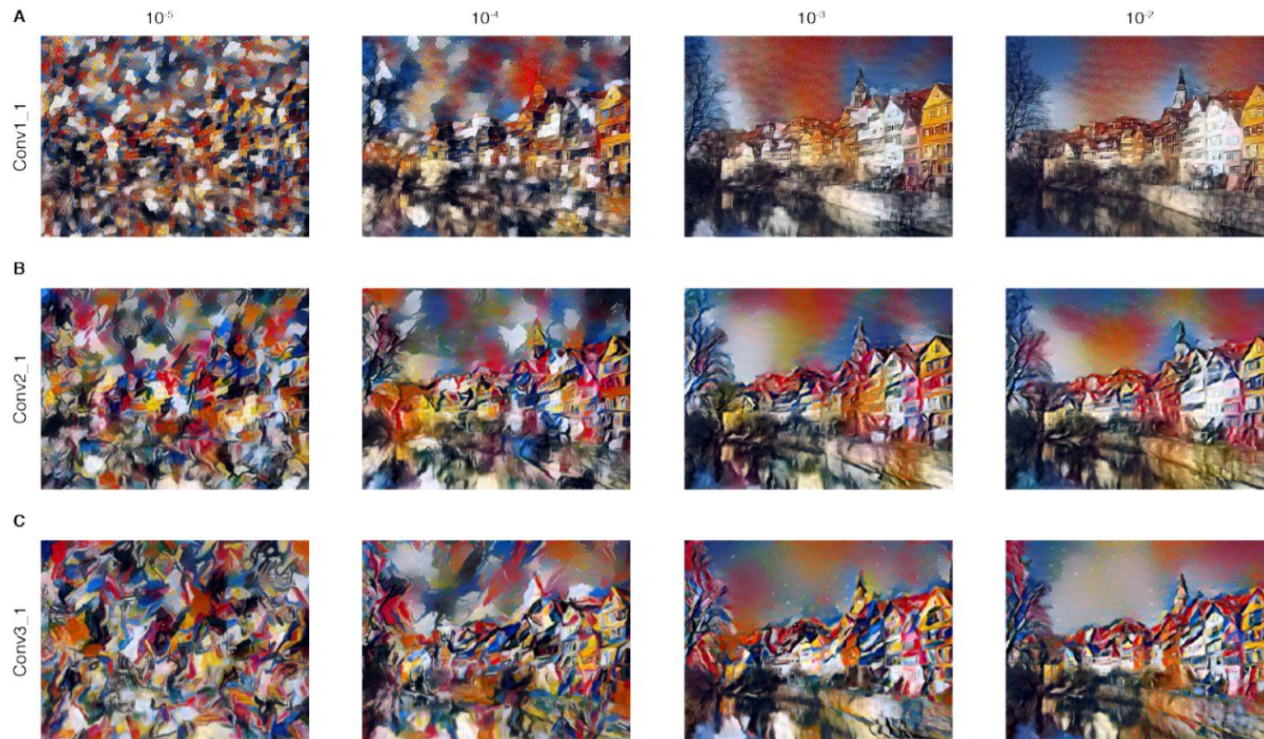
$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^L w_l E_l$$

$$\frac{\partial E_l}{\partial F_{ij}^l} = \begin{cases} \frac{1}{N_l^2 M_l^2} ((F^l)^T (G^l - A^l))_{ji} & \text{if } F_{ij}^l > 0 \\ 0 & \text{if } F_{ij}^l < 0 \end{cases}$$

# Реализация

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

# Реализация



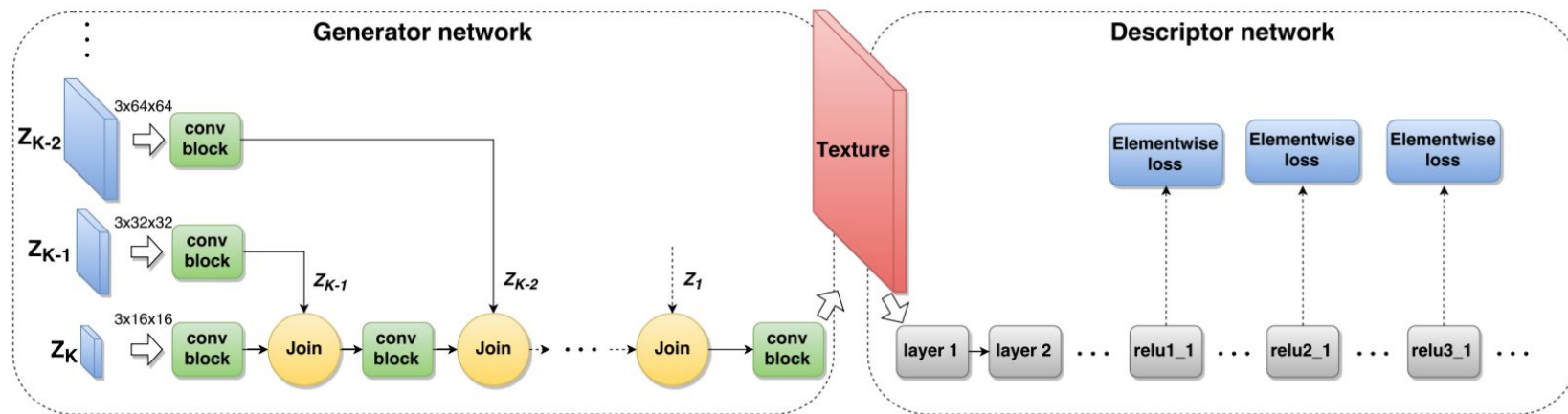
# Texture networks

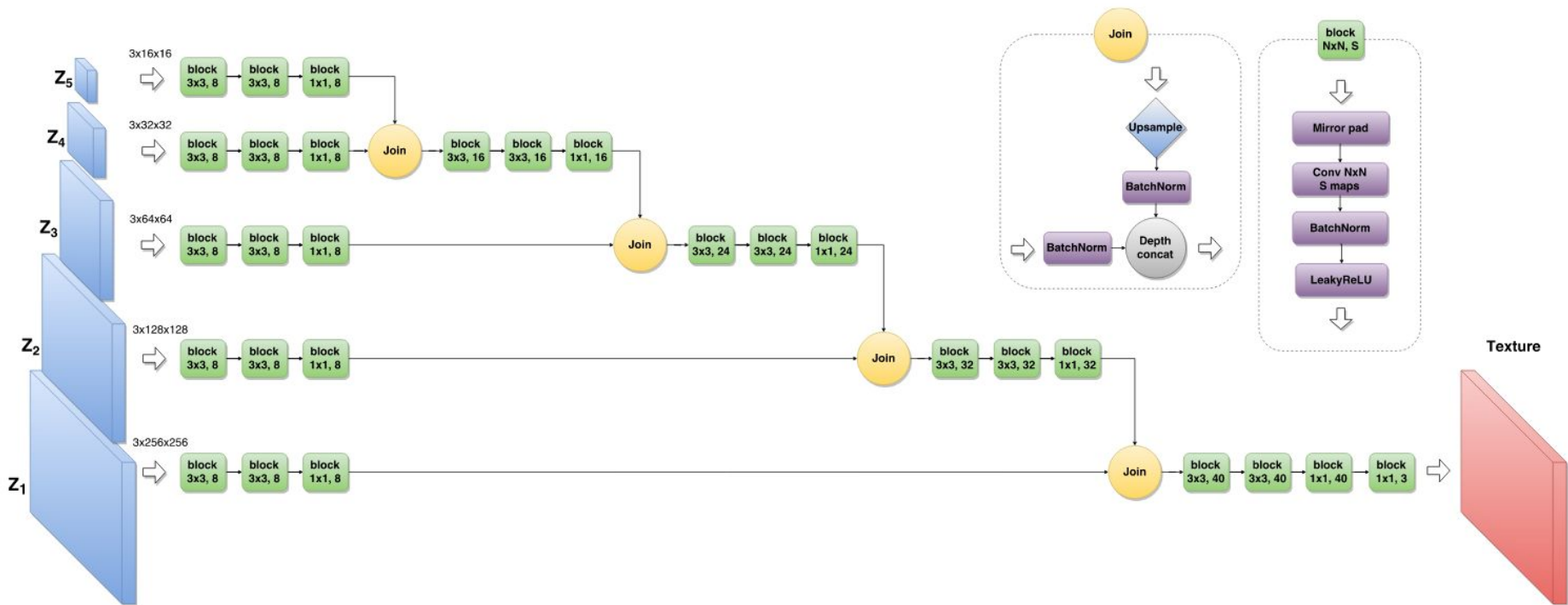
# Идея

Использовать ту же самую функцию потерь, что и в neural style

Но вместо градиентного спуска по белому шуму, обучить генератор

# Реализация





# Сравнение моделей



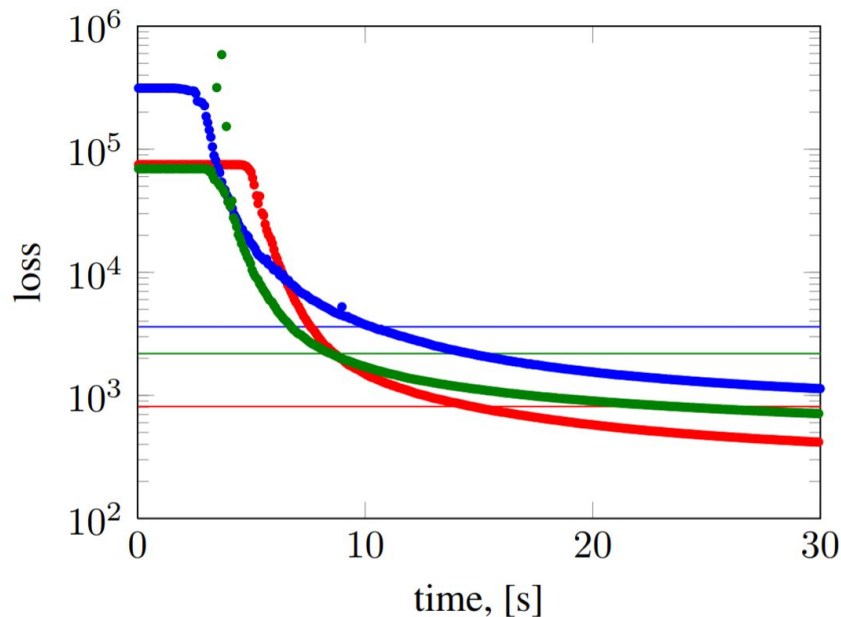
# Сравнение скорости

Толстые линии — neural style

Тонкие линии — texture networks

Neural style в итоге достигает большей точности, однако ему требуется около 10 секунд, чтобы достичь сравнимых с texture networks результатов.

Это в ~500 раз медленнее.



# Сравнение затрат памяти

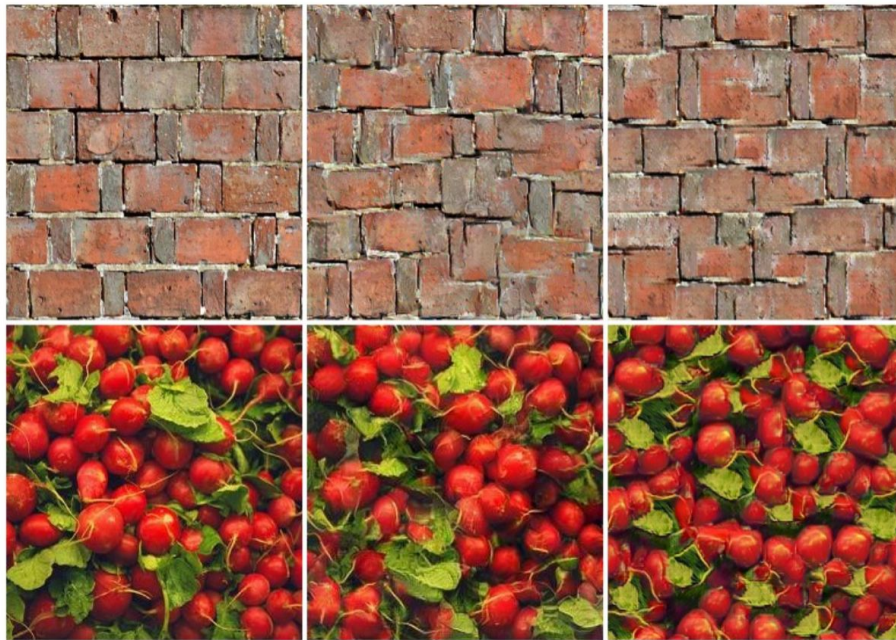
Texture networks не требуют backpropagation,  
поэтому занимают меньше памяти

Для создания текстуры 256 x 256

170 MB - texture networks

1100 MB - neural style

# Texture generation

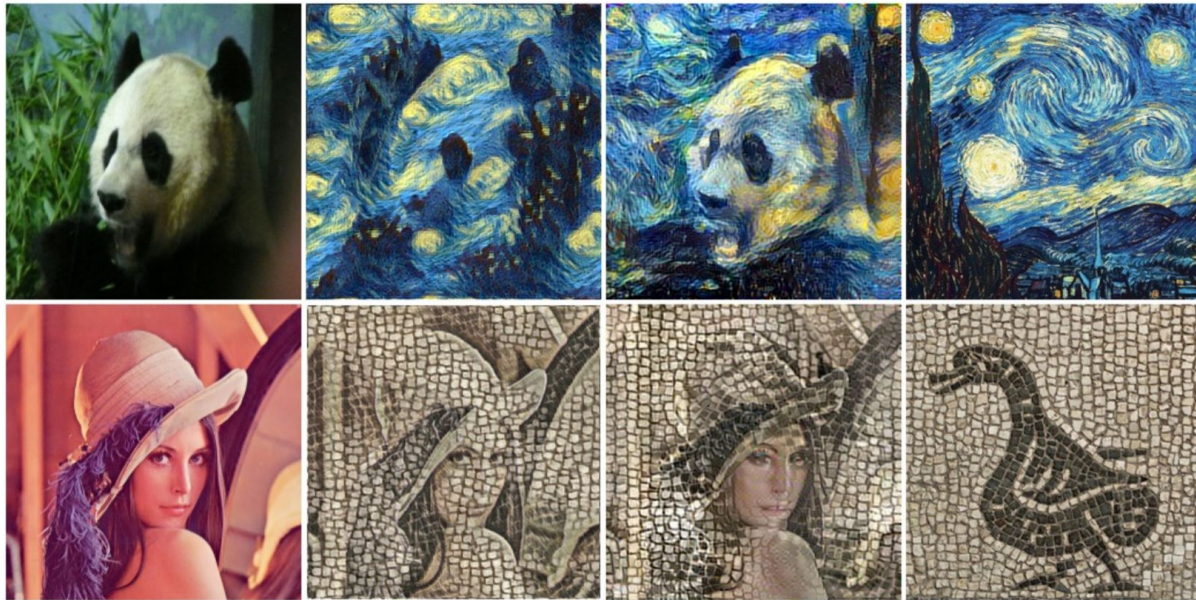


оригинал

Neural style

Texture networks

# Style transfer



оригинал

texture nets

neural style

стиль

# ИСТОЧНИКИ

- <https://arxiv.org/abs/1508.06576> — Neural style
- <https://arxiv.org/abs/1603.03417> — Texture networks