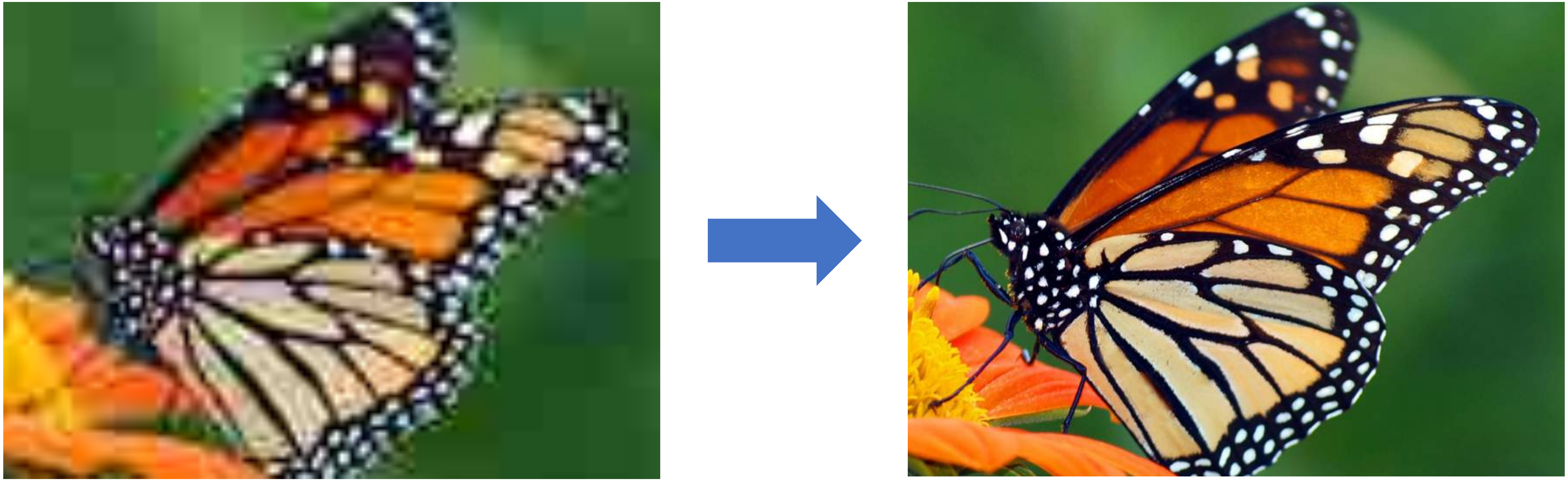
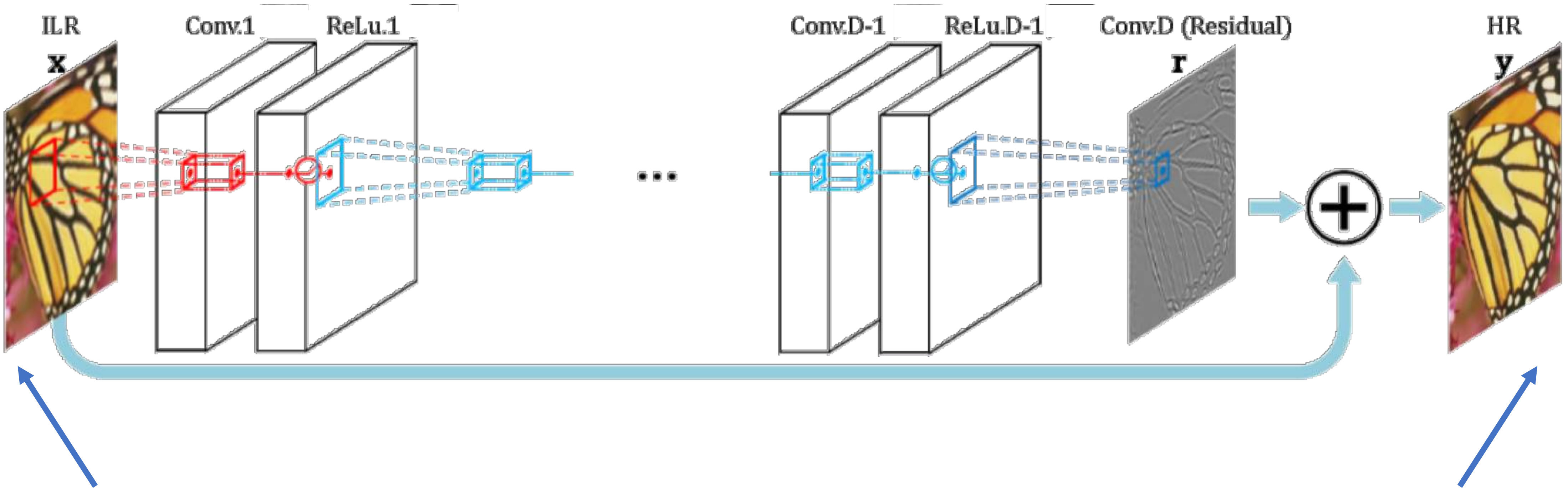
Generative Models

SuperResolution

SuperResolution



SuperResolution



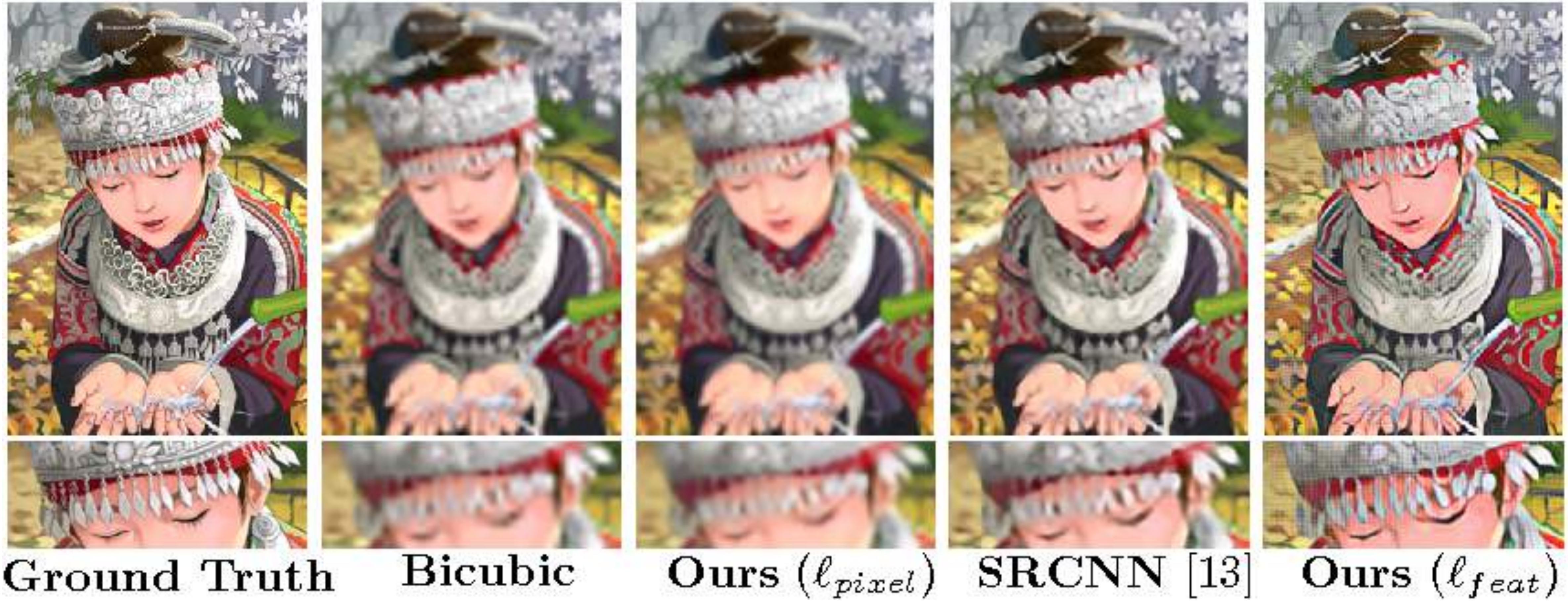
upsampled low-res image

$$\frac{1}{2}||\mathbf{y} - f(\mathbf{x})||^2$$

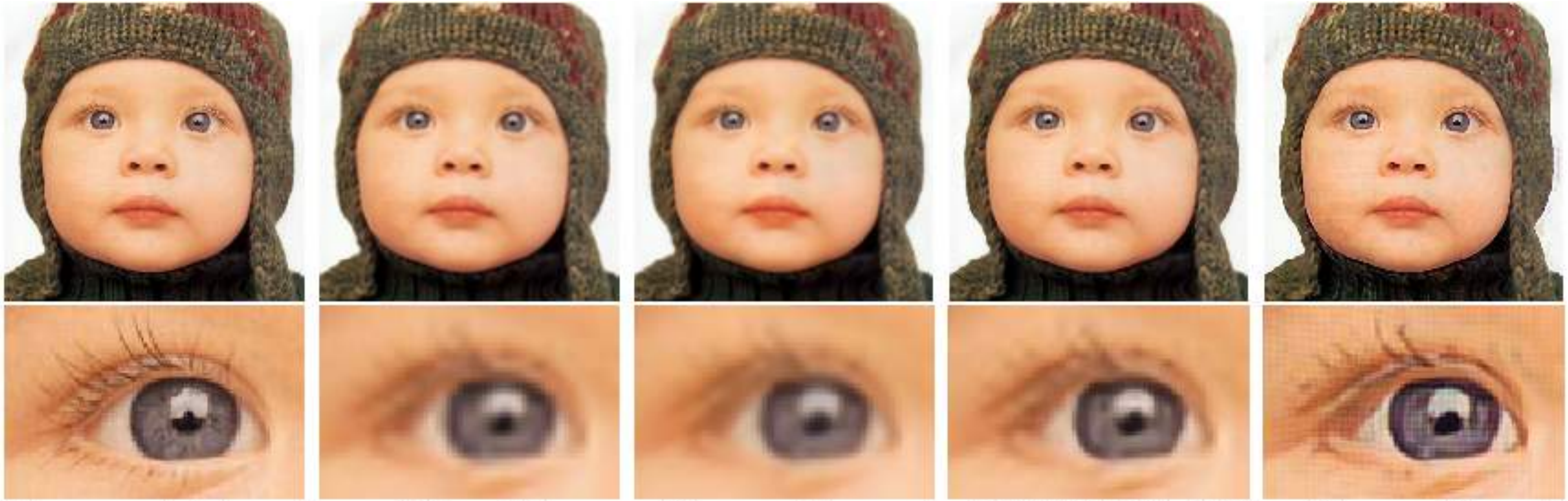
Приятная особенность задачи:
Не нужна разметка данных

Source: Kim et al. CVPR16

SuperResolution



SuperResolution



Ground Truth   Bicubic   Ours (ℓ_{pixel})   SRCNN [13]   Ours (ℓ_{feat})

Loss Functions

Pixel-wise MSE

$$\sum_i \|f(x_i;\theta) - y_i\|^2$$

В чем проблема?

Усреднение дает смазанную картинку

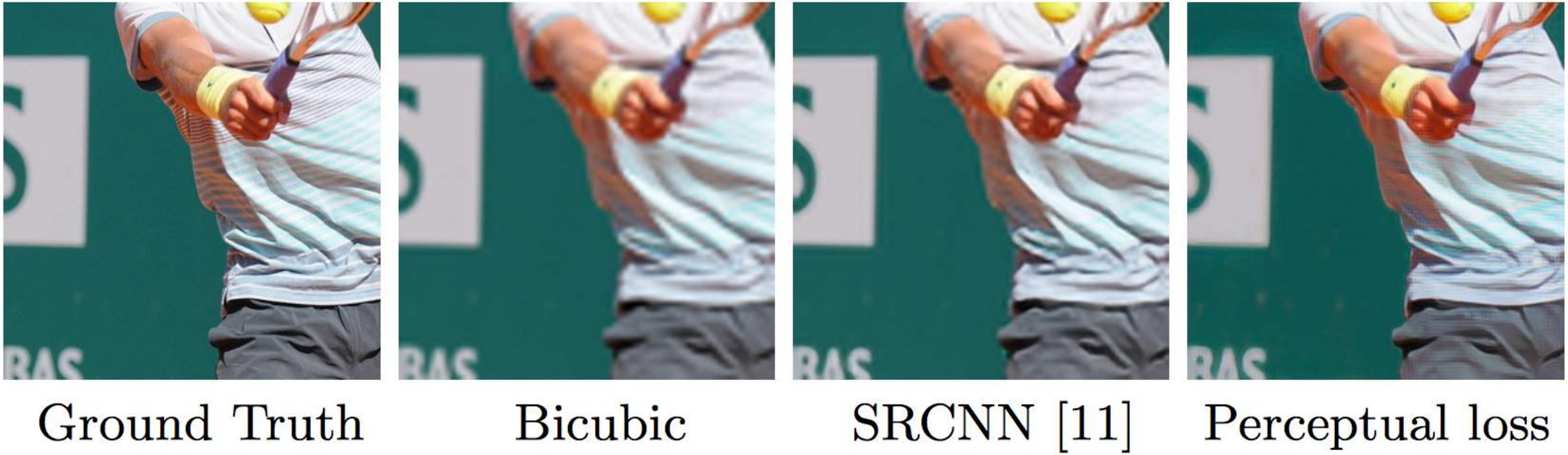
Perceptual loss

$$\frac{1}{C_j H_j W_j}\|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

Почему это хорошее решение?

Так мы можем ориентироваться на разрешение картинки в целом

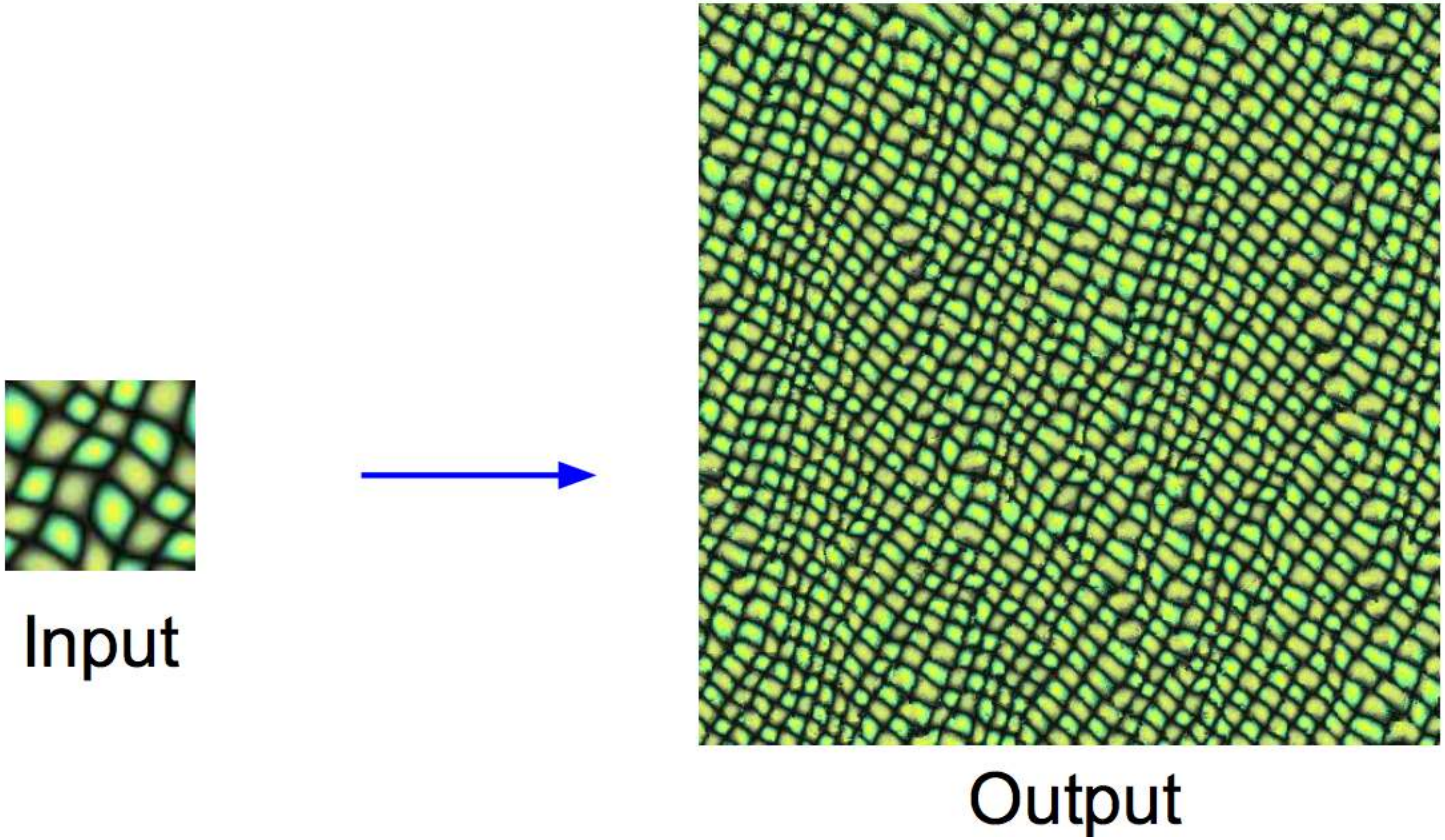
Loss Functions



Source: Johnson et al. CVPR16

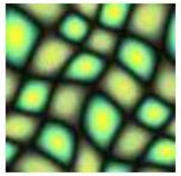
Texture Synthesis

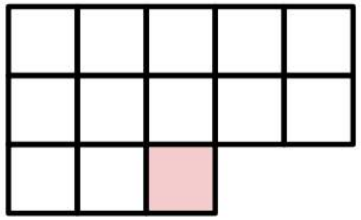
Texture Synthesis

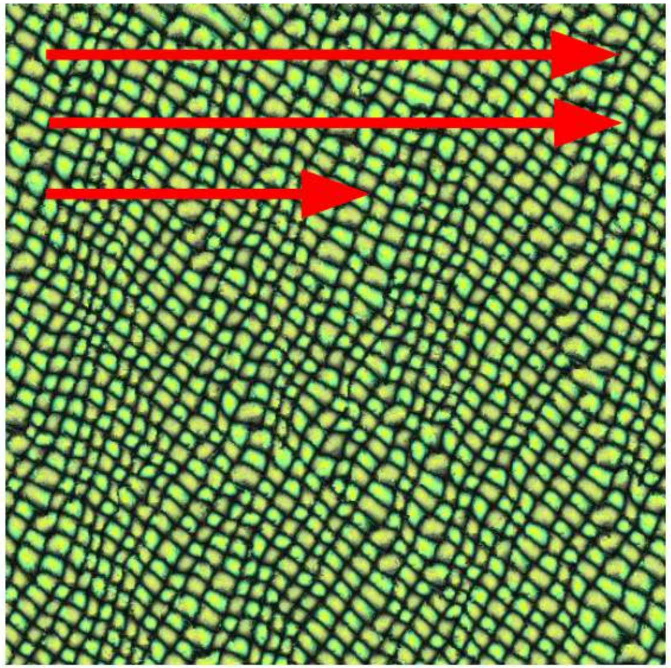


Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Texture Synthesis: Nearest Neighbor

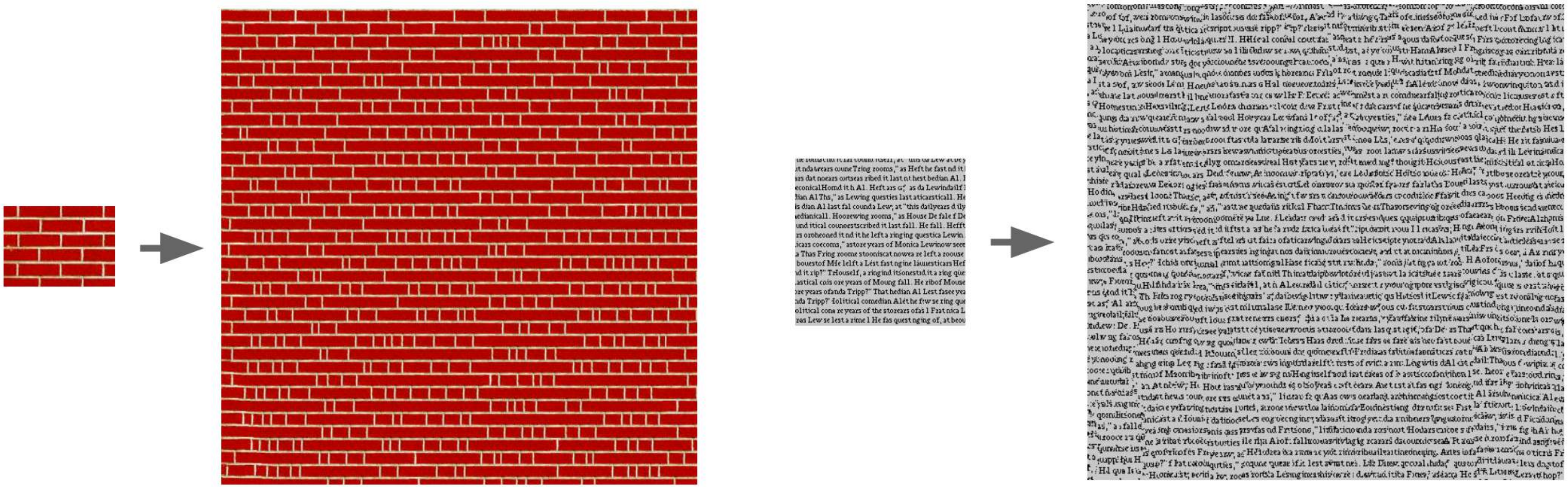






Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

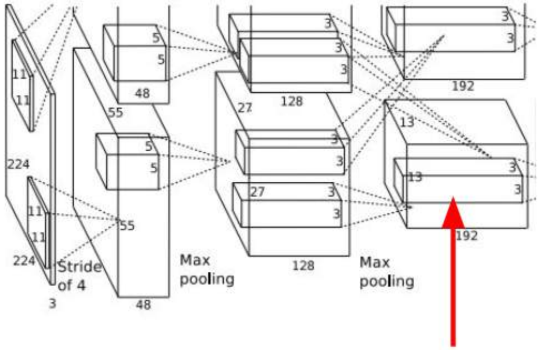
Texture Synthesis: Nearest Neighbor Results

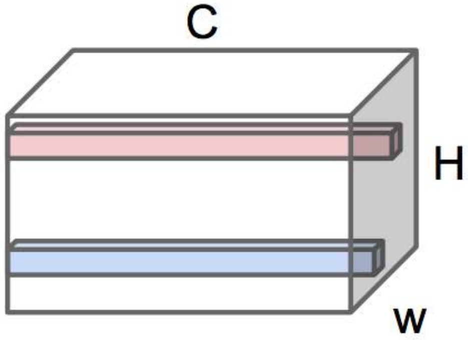


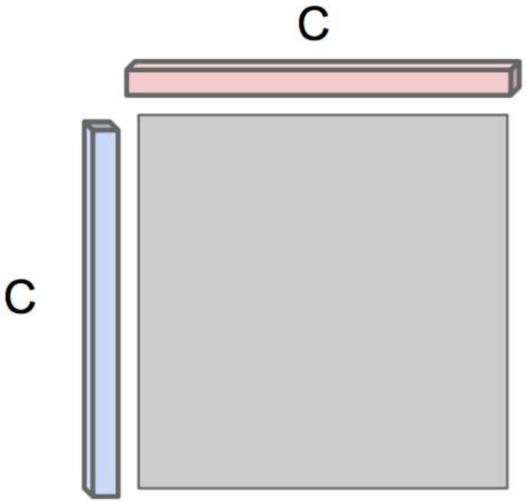
Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Texture Synthesis: Gram Matrix

This image is in the public domain.







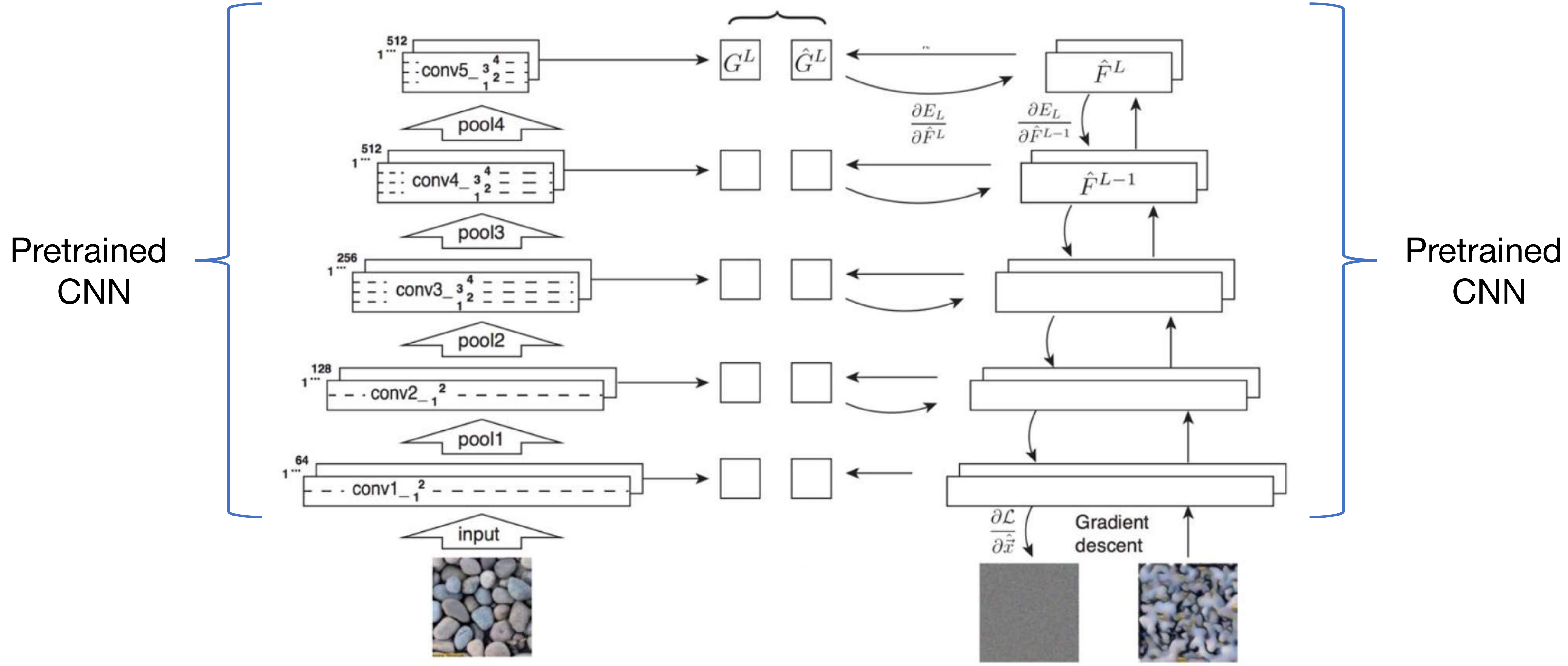
Каждый слой CNN дает тензор C x H x W или сетку H x W векторов размерности C.

Внешнее произведение двух C-мерных векторов дает матрицу C x C, которая измеряет совстречаемость паттернов.

Усредним значения всех возможных матриц и получим матрицу Грама.

Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

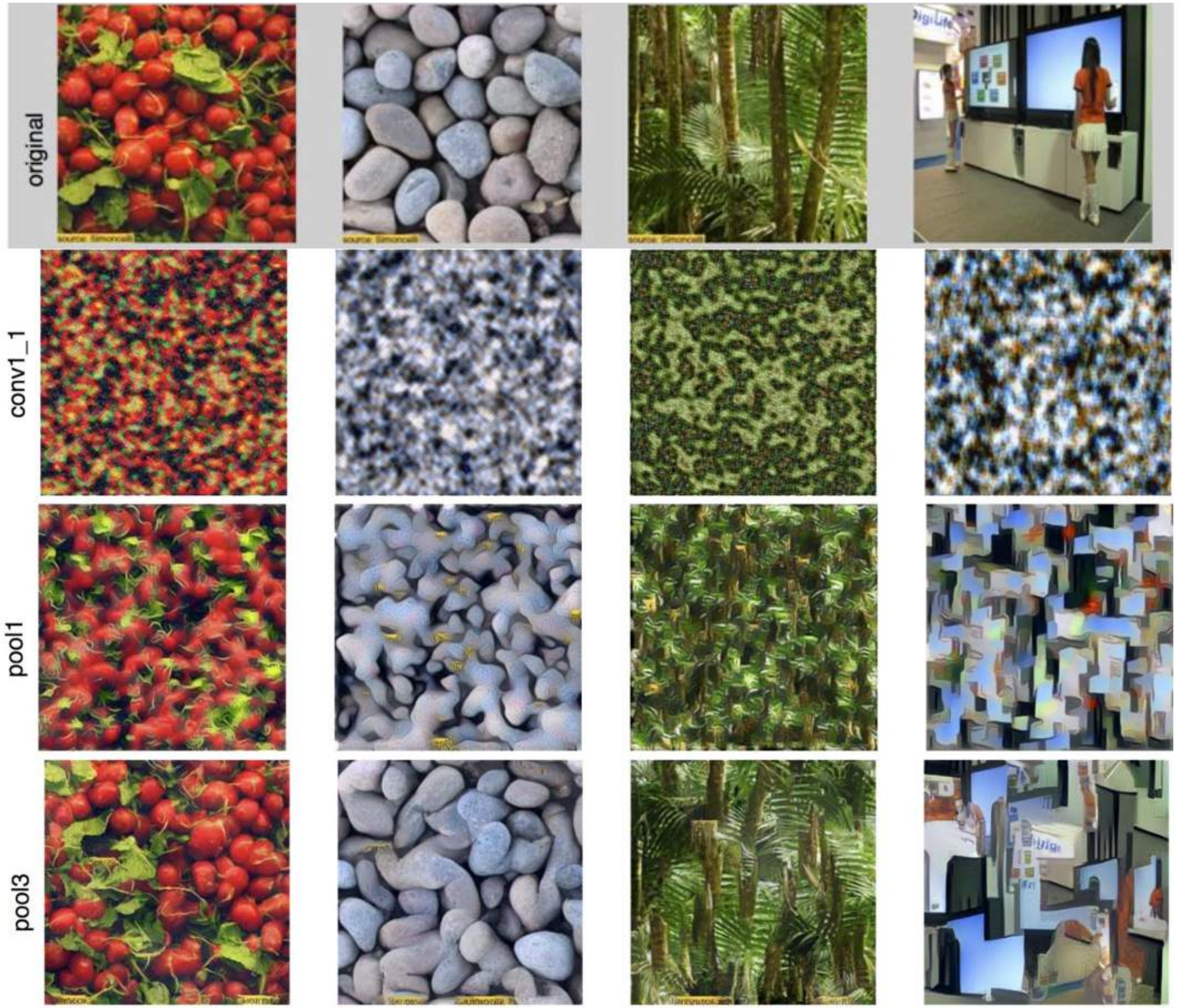
Neural Texture Synthesis



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

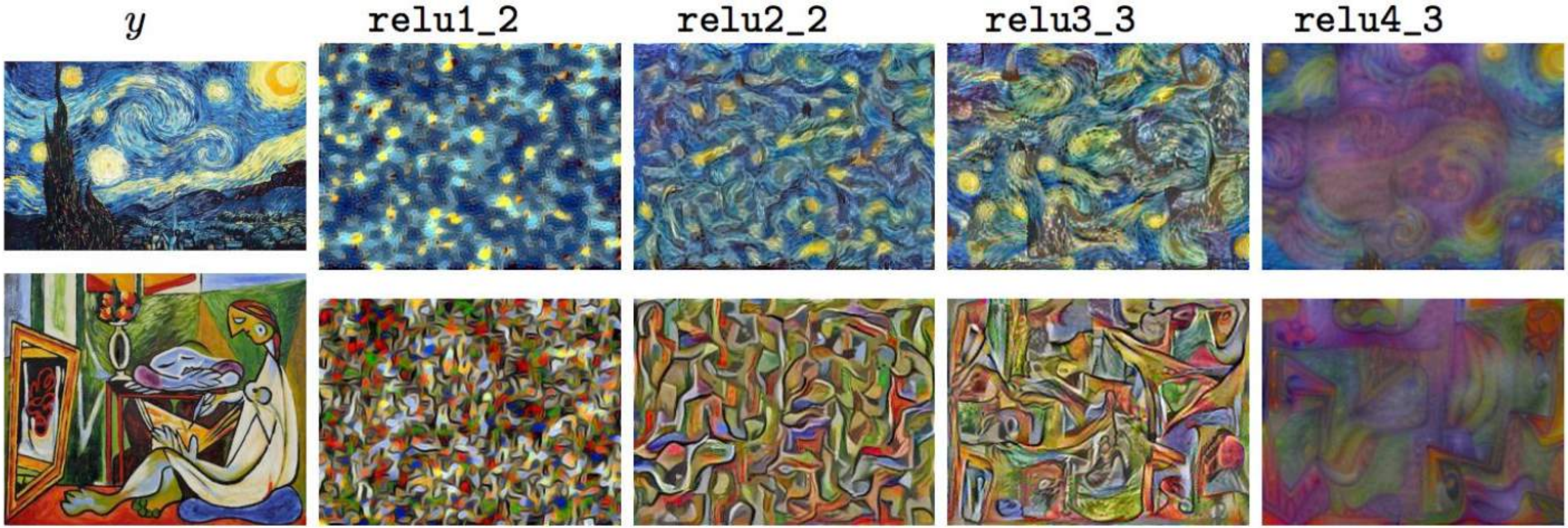
Neural Texture Synthesis

Реконструкция изображений с верхних слоев сети позволяет экстрактировать большие признаки из оригинального изображения.



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Neural Texture Synthesis

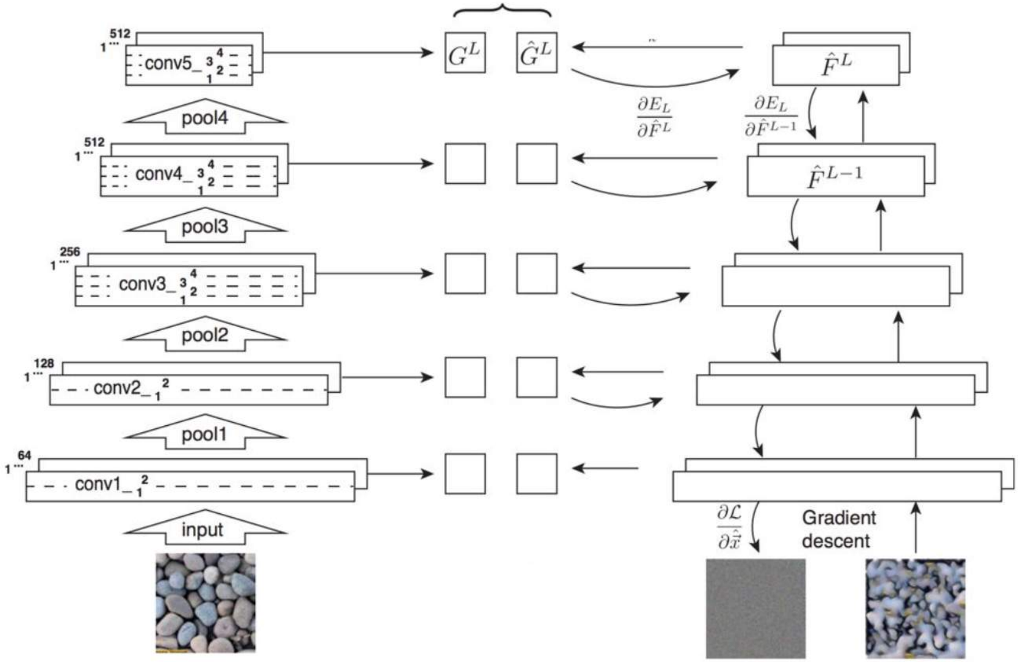


Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Perceptual Loss + Texture Synthesis

Perceptual loss

$$\frac{1}{C_j H_j W_j}\|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

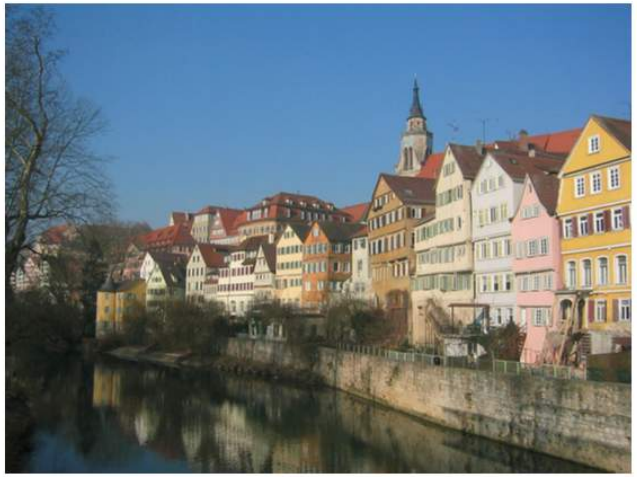

512
conv5_
pool4
512
conv4_
pool3
256
conv3_
pool2
128
conv2_
pool1
64
conv1_
input
G^L
$\hat{G}^L$
$\hat{F}^L$
$\frac{\partial E_L}{\partial \hat{F}^L}$
$\frac{\partial E_L}{\partial \hat{F}^{L-1}}$
$\hat{F}^{L-1}$
$\frac{\partial \mathcal{L}}{\partial \hat{\vec{x}}}$
Gradient descent

?

Style Transfer

Style Transfer

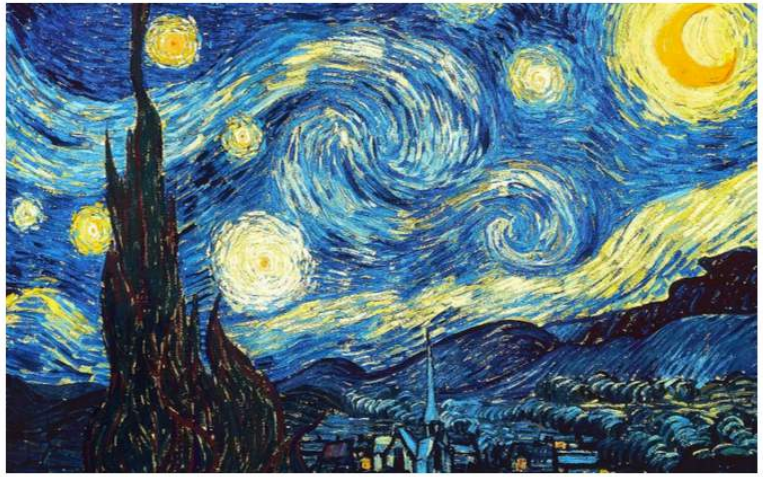
Content Image



This image is licensed under CC-BY 3.0

+

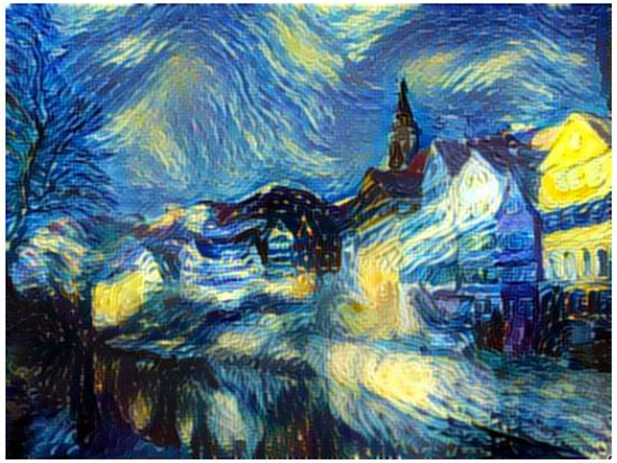
Style Image



Starry Night by Van Gogh is in the public domain

=

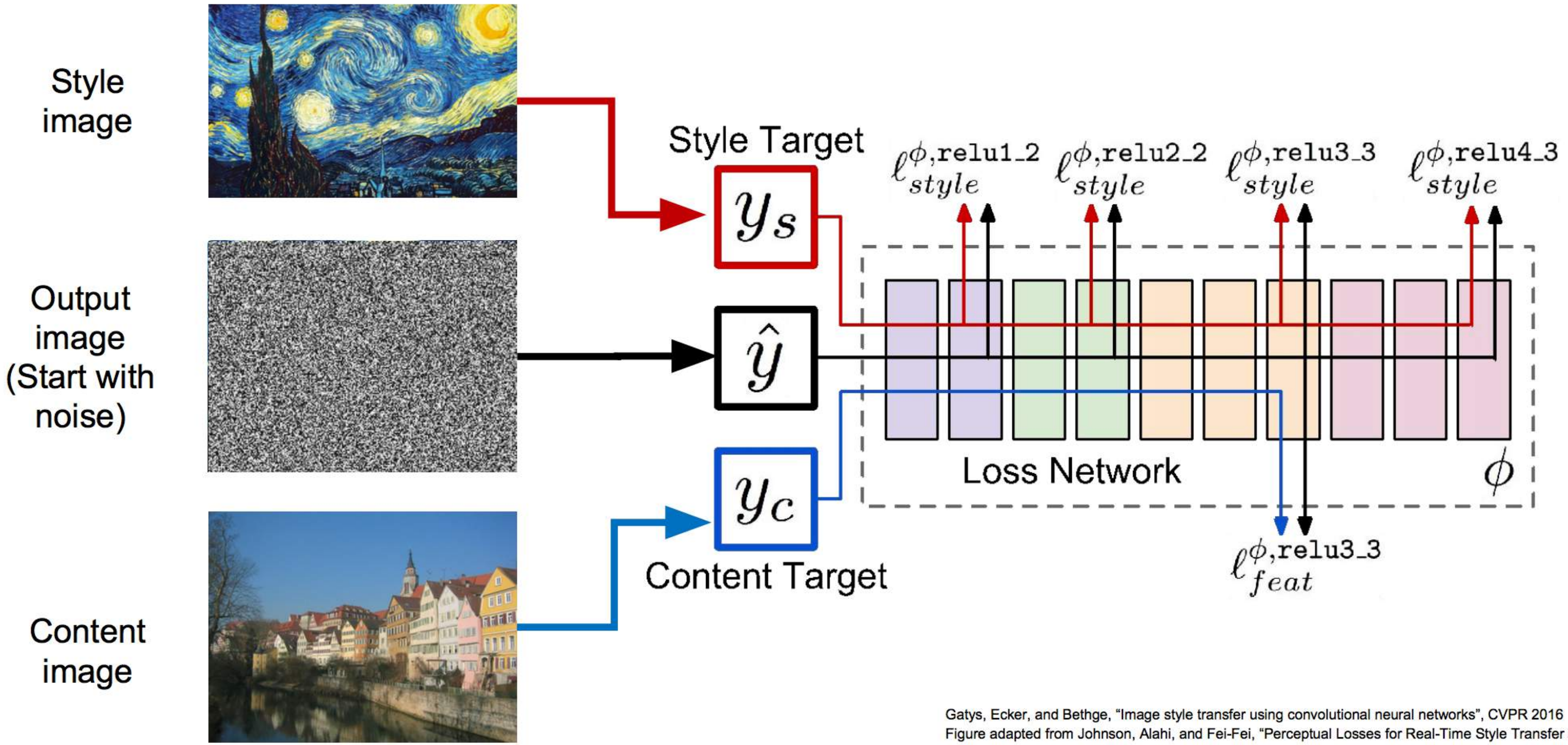
Style Transfer!



This image copyright Justin Johnson, 2015. Reproduced with permission.

Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

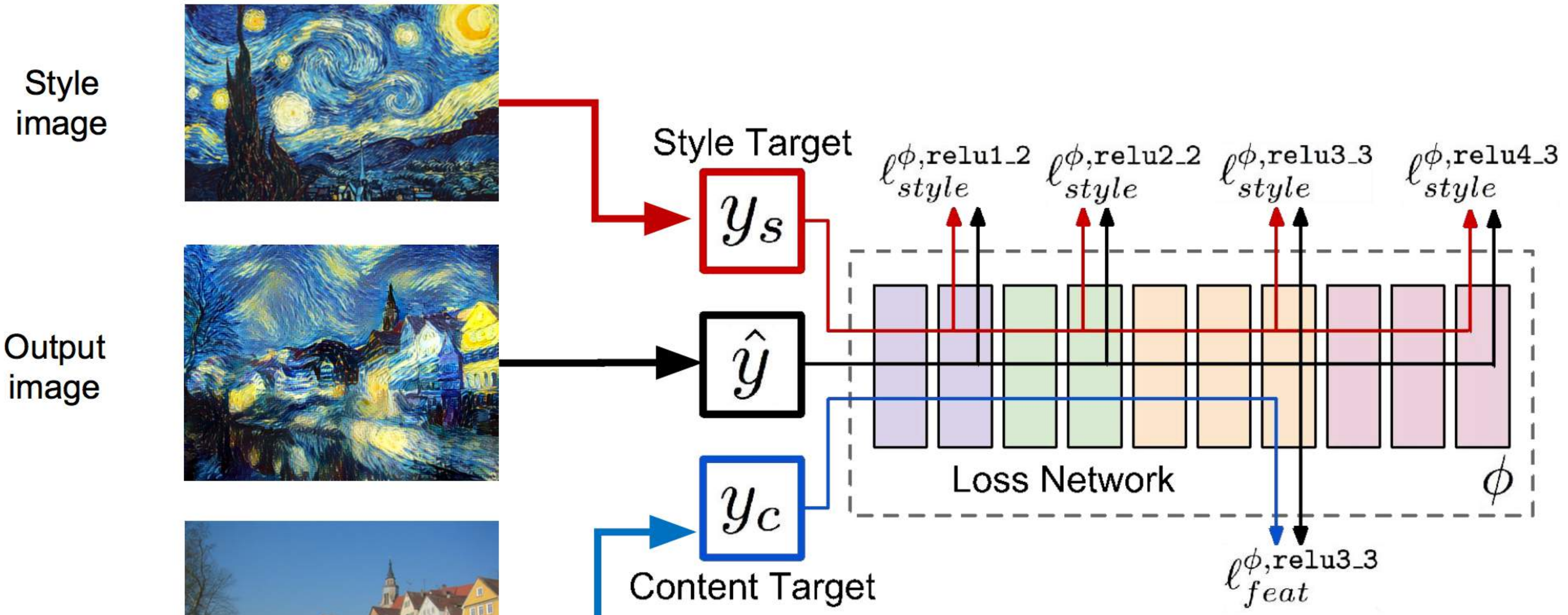
Style Transfer


Style image
Output image (Start with noise)
Content image
Style Target
y_s
$\hat{y}$
y_c
Content Target
$\ell_{style}^{\phi,\texttt{relu1\_2}}$
$\ell_{style}^{\phi,\texttt{relu2\_2}}$
$\ell_{style}^{\phi,\texttt{relu3\_3}}$
$\ell_{style}^{\phi,\texttt{relu4\_3}}$
Loss Network
ϕ
$\ell_{feat}^{\phi,\texttt{relu3\_3}}$

Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016
Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

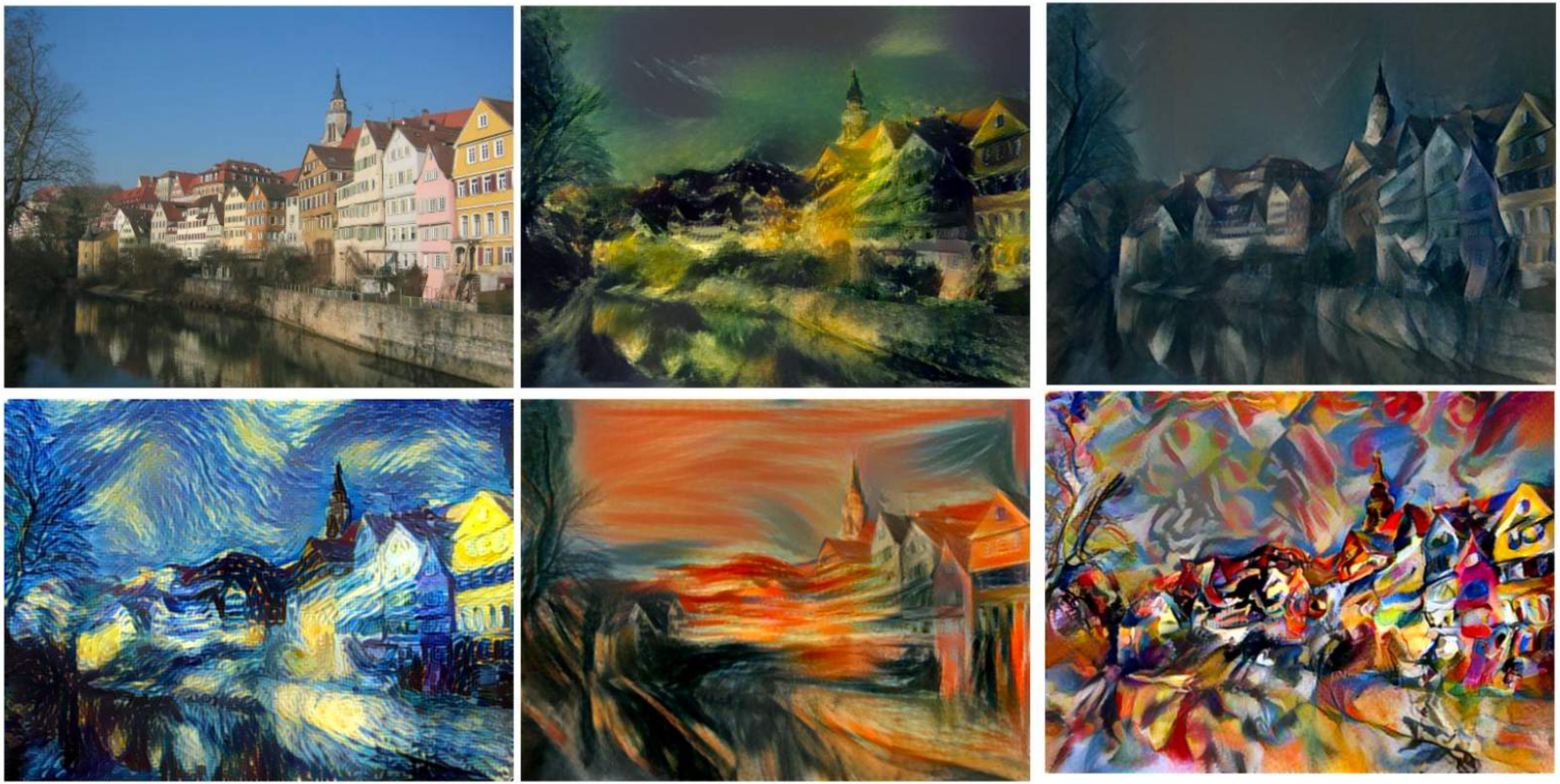
Style Transfer


Style image
Output image
Content image
Style Target
y_s
$\hat{y}$
y_c
Content Target
$\ell_{style}^{\phi,\mathtt{relu1\_2}}$
$\ell_{style}^{\phi,\mathtt{relu2\_2}}$
$\ell_{style}^{\phi,\mathtt{relu3\_3}}$
$\ell_{style}^{\phi,\mathtt{relu4\_3}}$
Loss Network
ϕ
$\ell_{feat}^{\phi,\mathtt{relu3\_3}}$

Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016
Figure adapted from Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016. Copyright Springer, 2016. Reproduced for educational purposes.

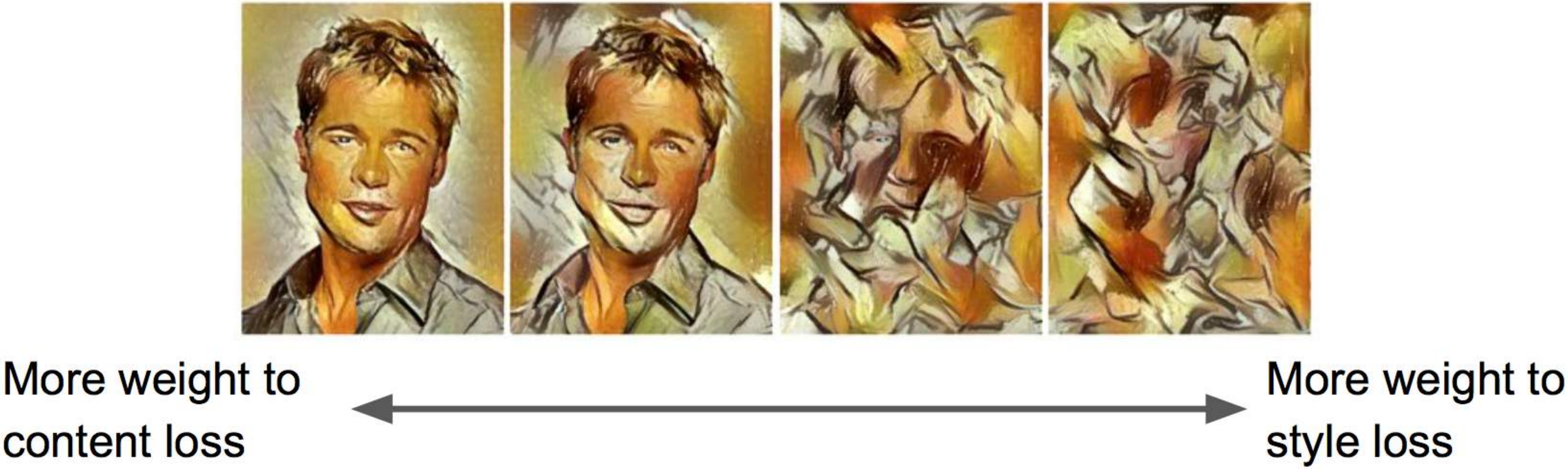
Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Style Transfer



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Style Transfer

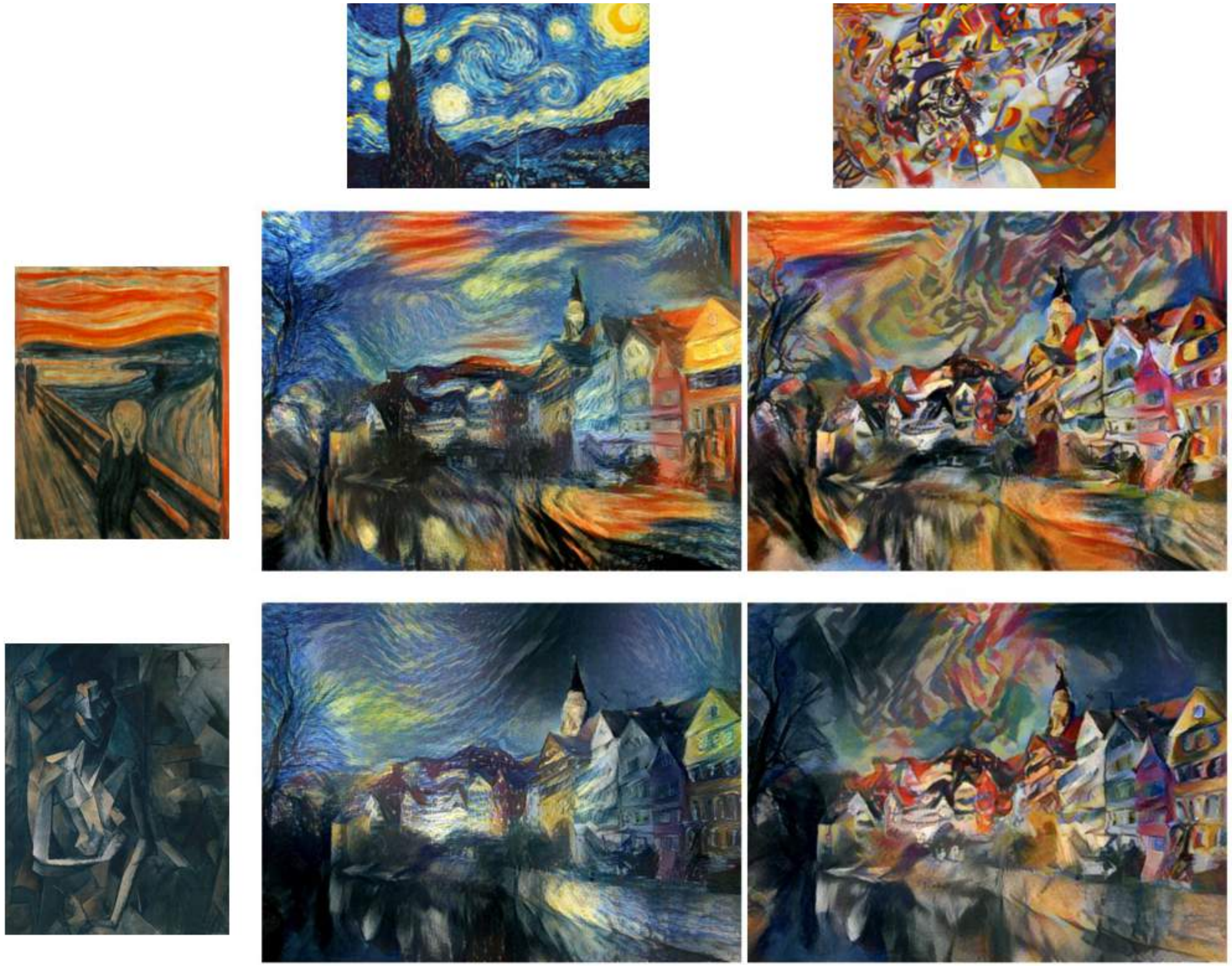


More weight to content loss

More weight to style loss

Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Style Transfer: Mixing Styles



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Style Transfer

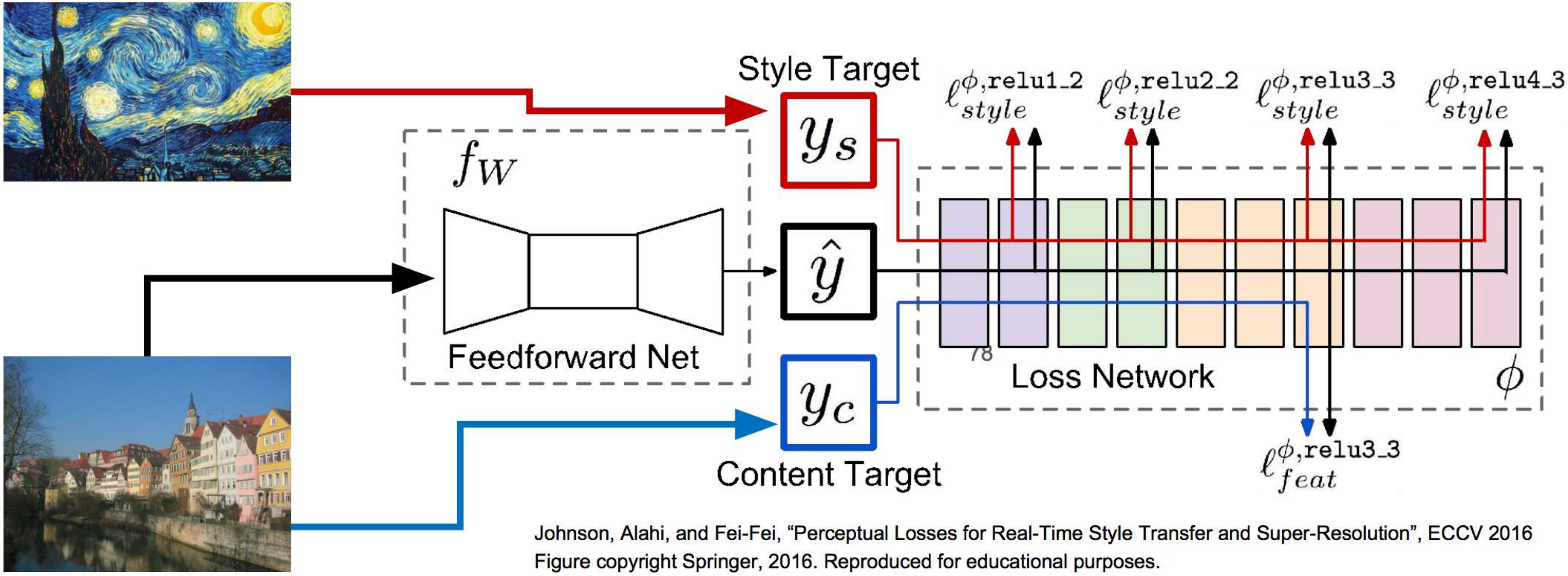
Проблема:
В процессе оптимизации требуется много раз пройти через нейронную сеть, чтобы получить изображение. Это долго.

Решение:
Натренировать еще одну нейронную сеть делать style transfer (имитировать процесс оптимизации).

Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Style Transfer

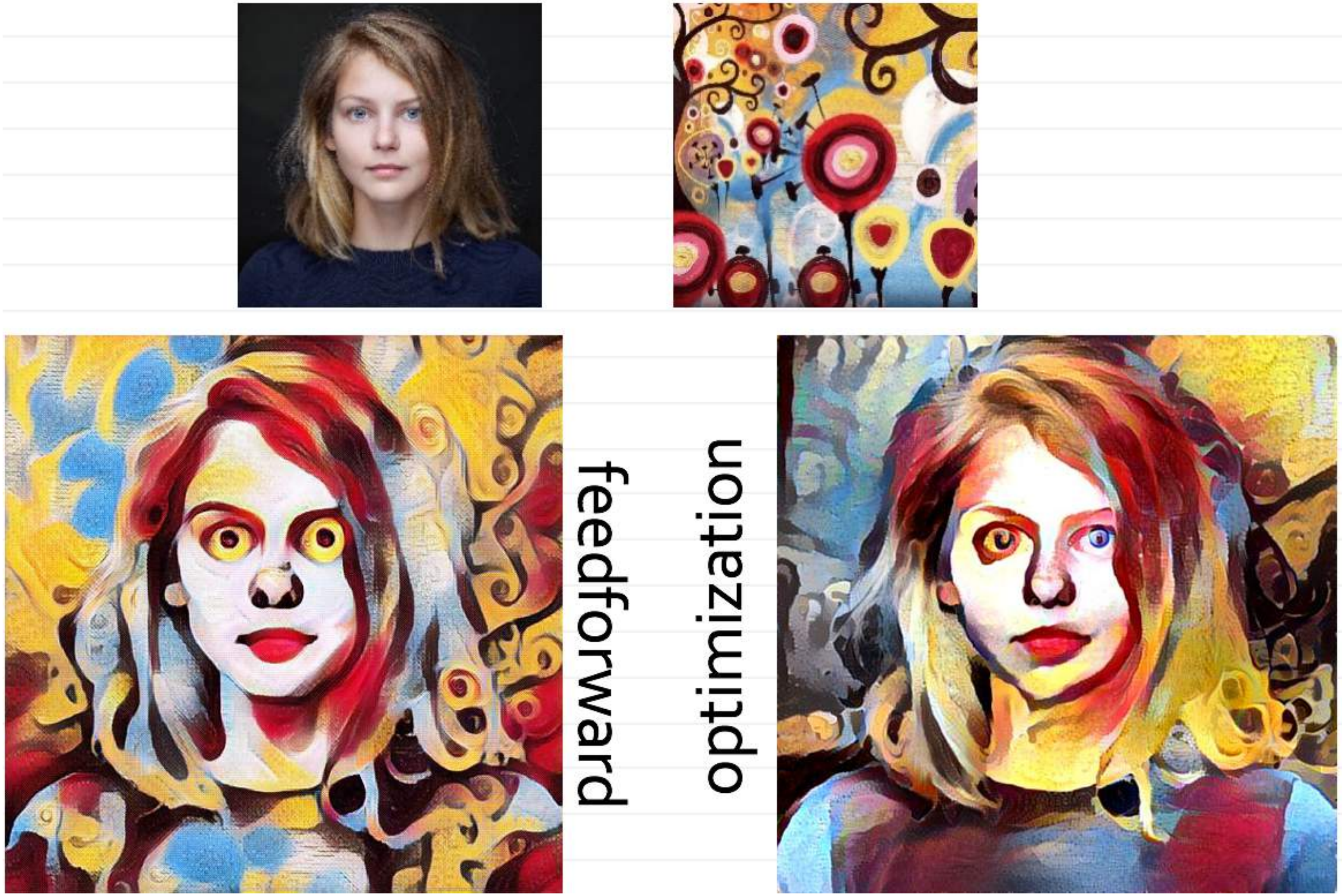
Для каждого стиля тренируем по одной нейронной сети.


Style Target
y_s
f_W
$\hat{y}$
Feedforward Net
y_c
Content Target
$\ell_{style}^{\phi,\text{relu1\_2}}$
$\ell_{style}^{\phi,\text{relu2\_2}}$
$\ell_{style}^{\phi,\text{relu3\_3}}$
$\ell_{style}^{\phi,\text{relu4\_3}}$
Loss Network
ϕ
$\ell_{feat}^{\phi,\text{relu3\_3}}$
78

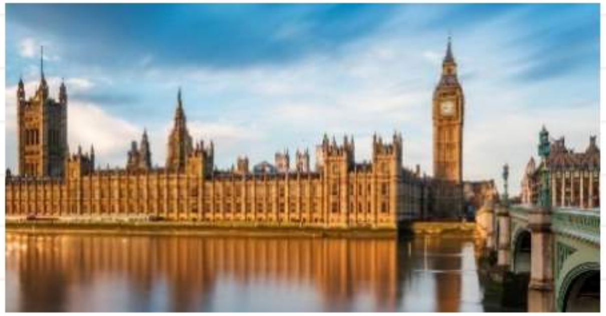
Johnson, Alahi, and Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", ECCV 2016
Figure copyright Springer, 2016. Reproduced for educational purposes.

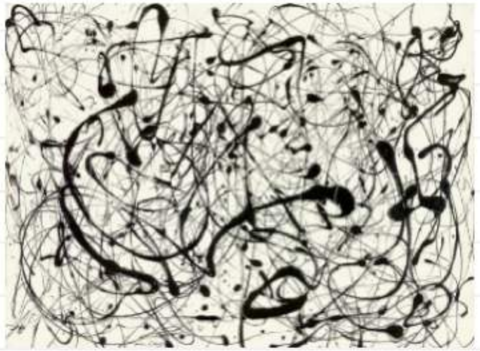
Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

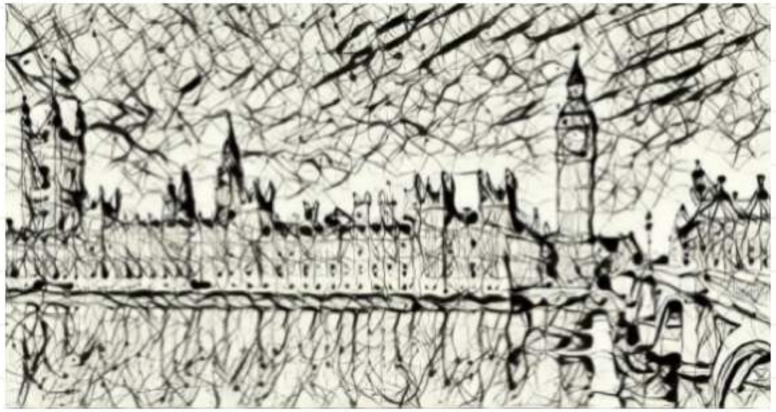
Style Transfer



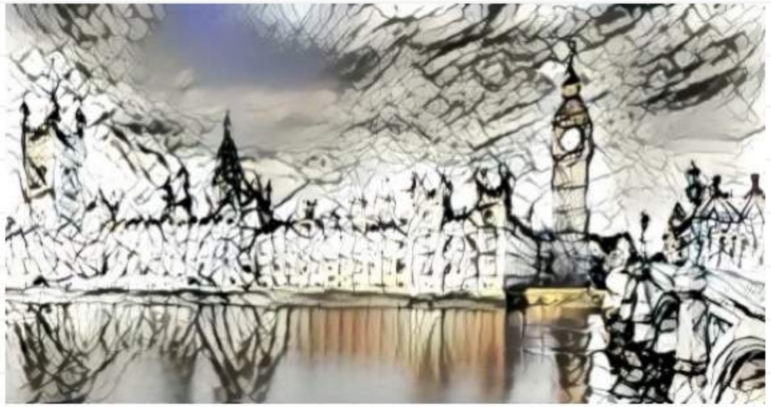
Style Transfer





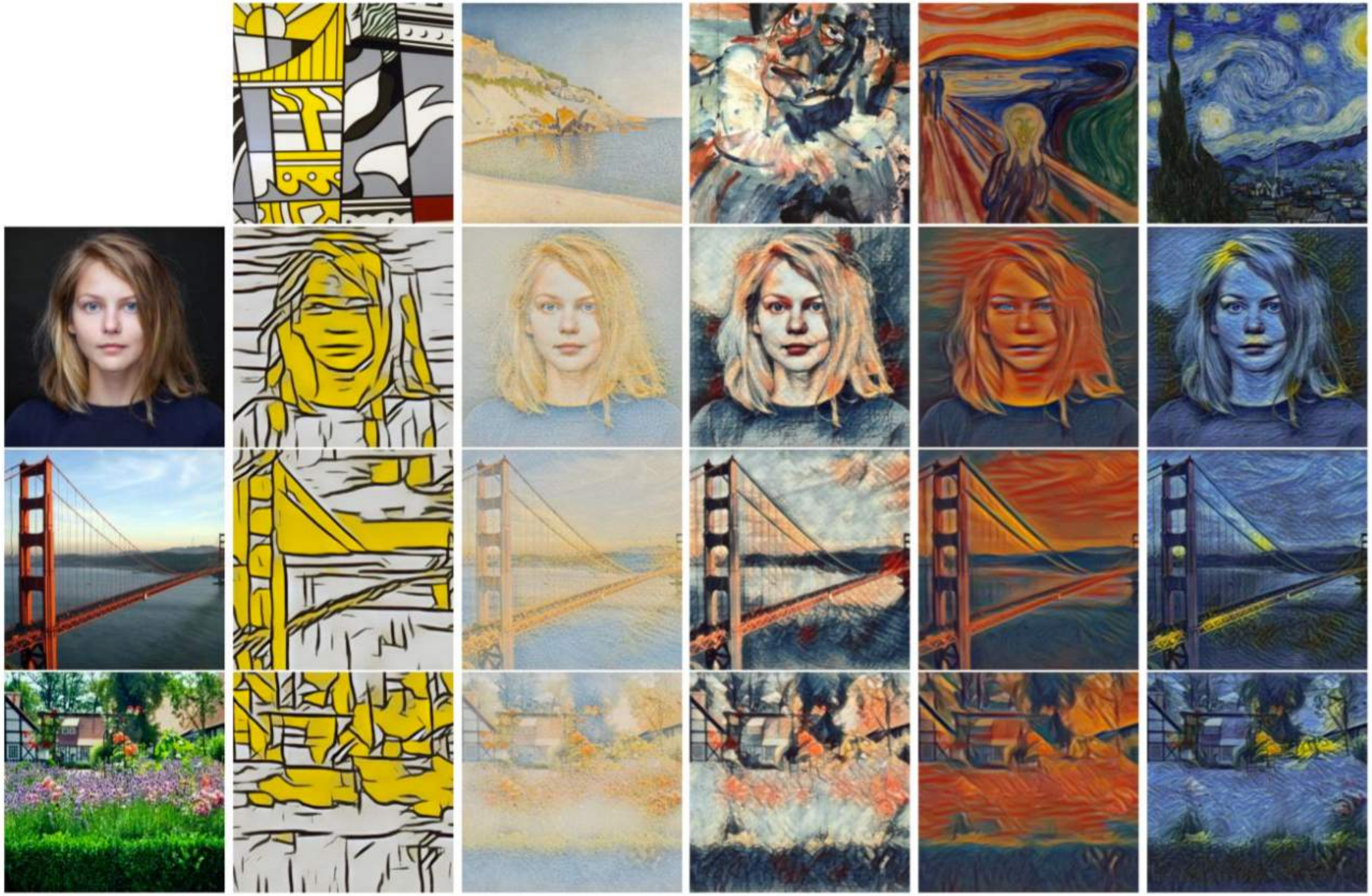


feedforward



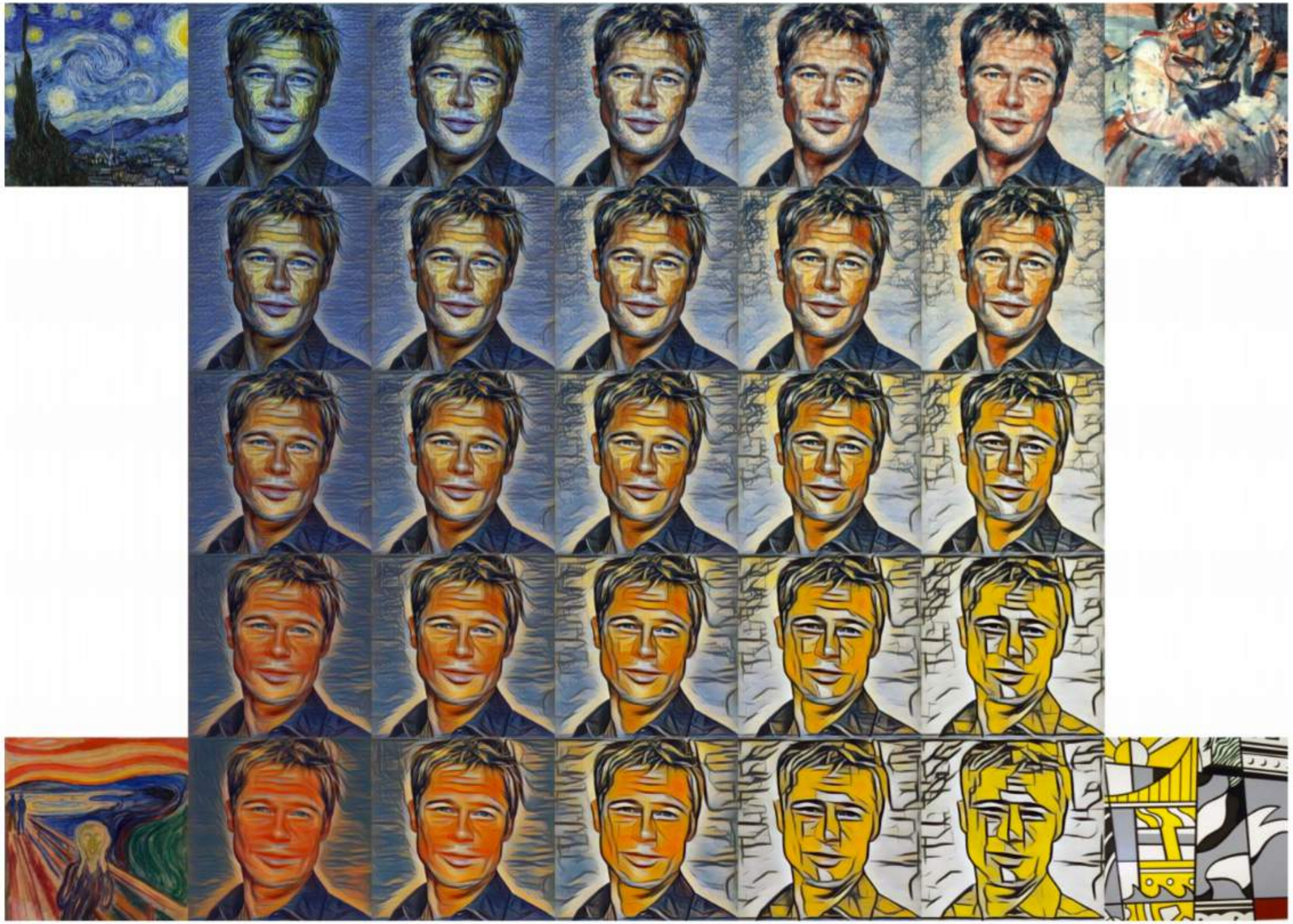
optimization

Style Transfer: One Net Many Styles



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

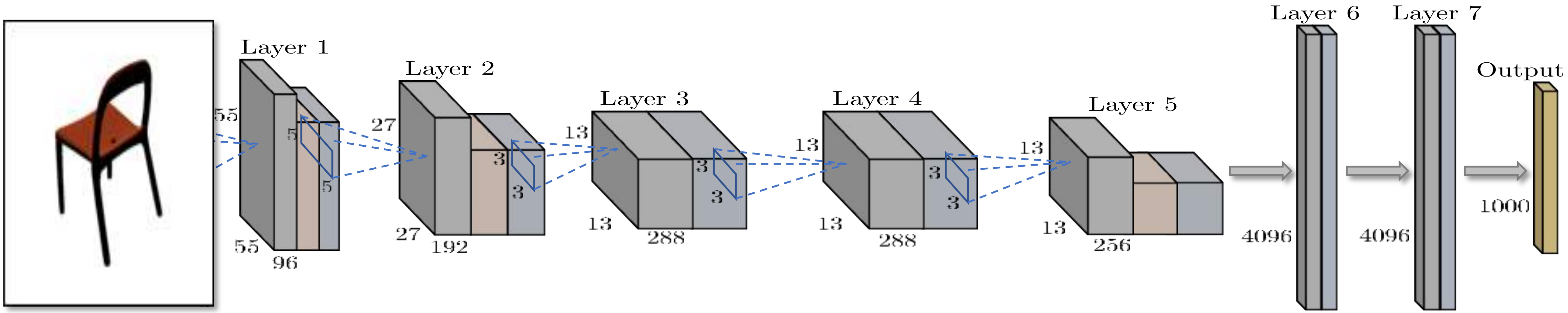
Style Transfer: Morphing Between Styles



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture12.pdf

Conditional Generation

Conditional Generation

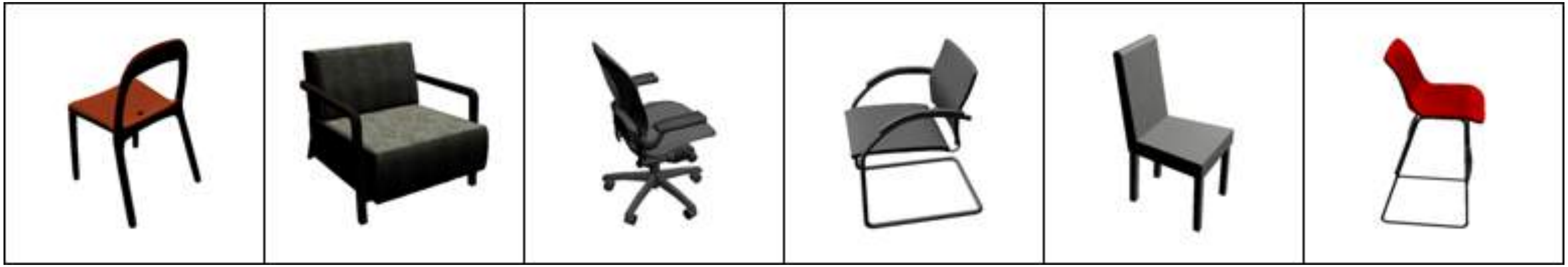


У нас есть картинки стульев,

мы хотим их классифицировать на разные модели.

Source: Dosovitskiy et al. CVPR 2015

Conditional Generation

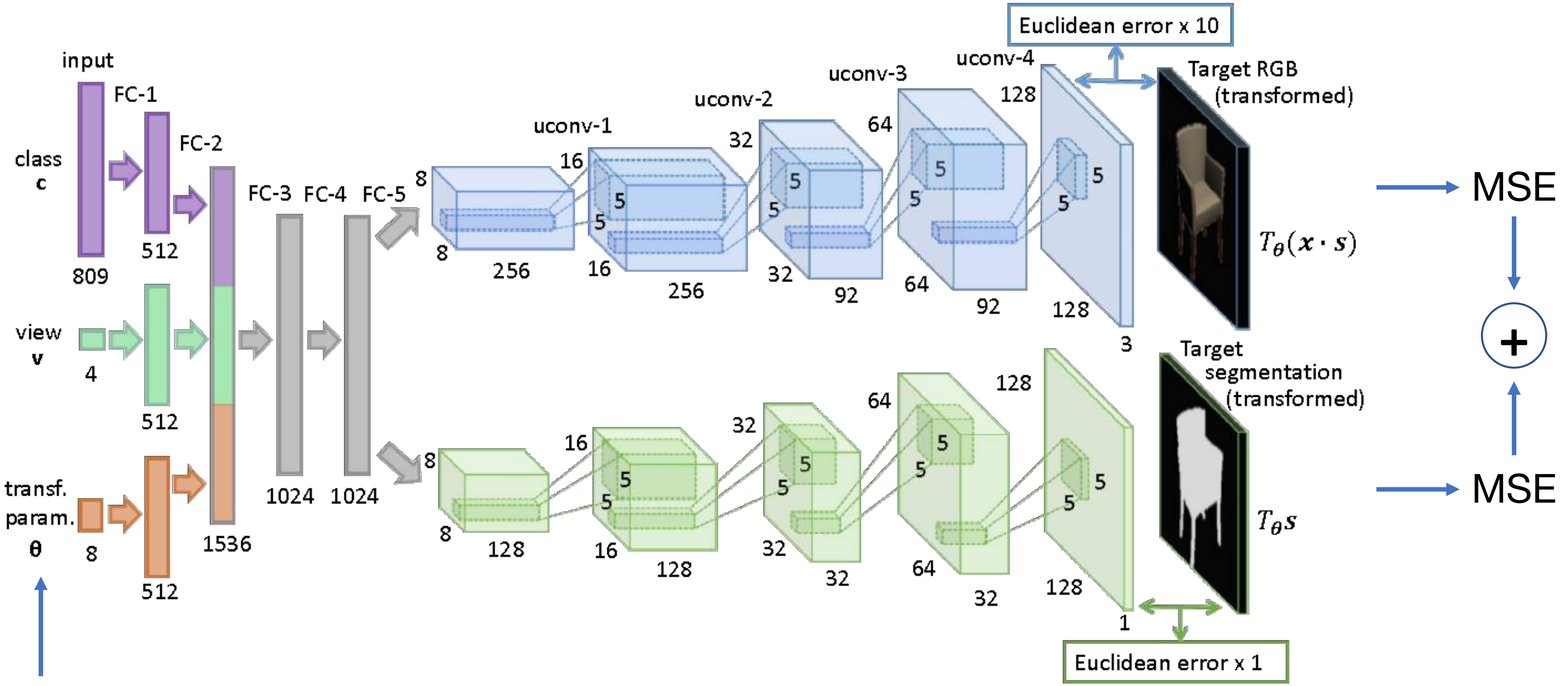


Для каждой картинки у нас есть ID стула и параметры угла обзора.

Мы хотим генерировать стулья исходя из этих признаков.

Source: Dosovitskiy et al. CVPR 2015

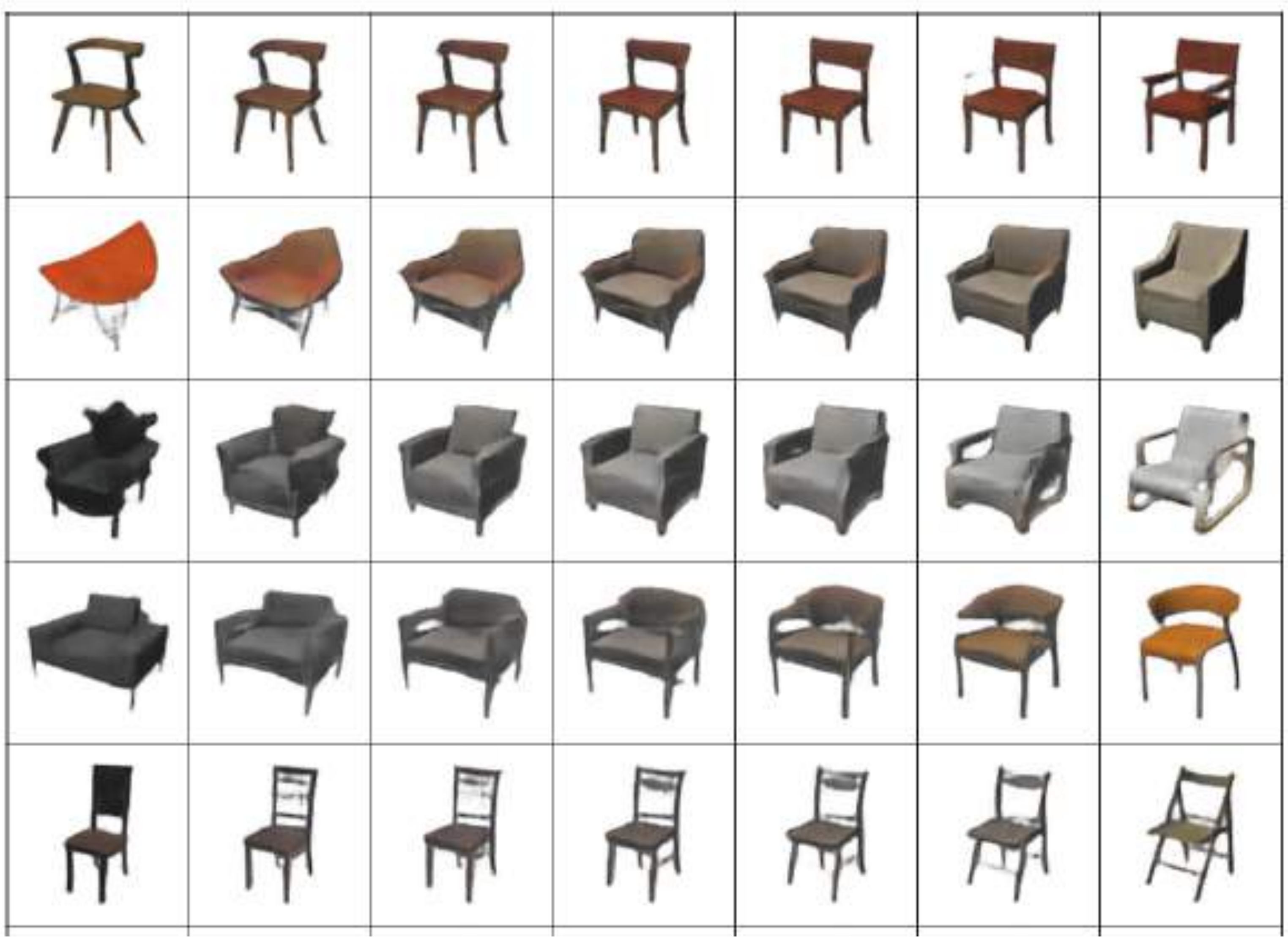
Conditional Generation



in-plane rotation, translation, zoom, stretching horizontally or vertically etc.

Source: Dosovitskiy et al. CVPR 2015

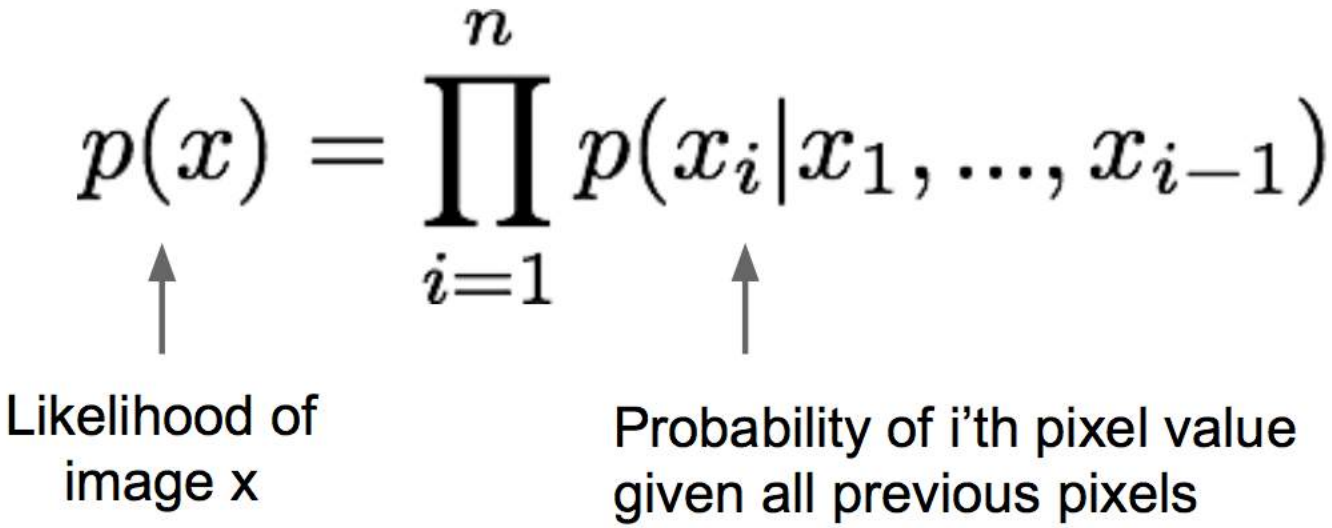
Conditional Generation: Morphing Chairs



Source: Dosovitskiy et al. CVPR 2015

PixelRNN and PixelCNN

Fully Visible Belief Network



$$p(x) = \prod_{i=1}^{n} p(x_i | x_1, ..., x_{i-1})$$

Likelihood of image x

Probability of i’th pixel value given all previous pixels

Как моделировать распределения пикселей?

Какой-нибудь нейронной сетью.

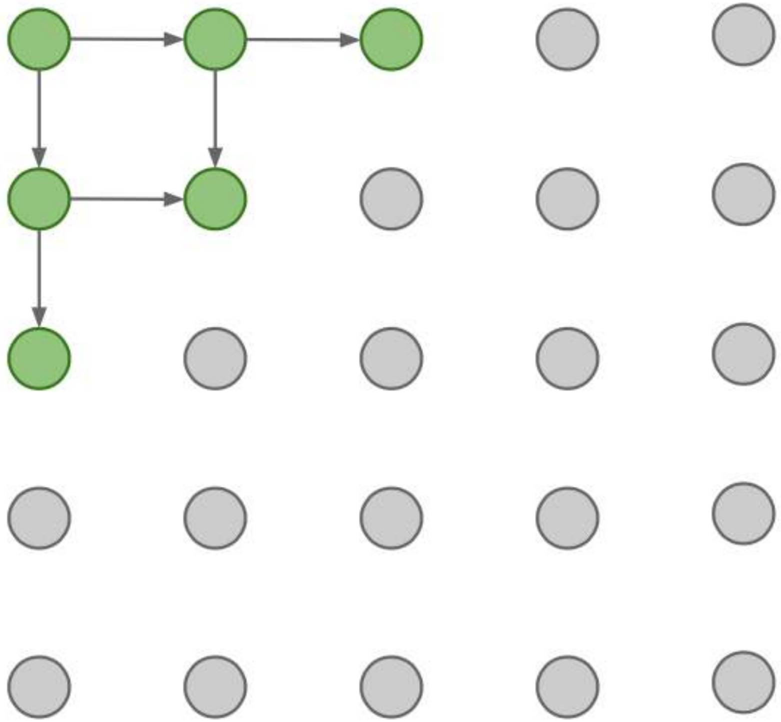
Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

PixelRNN

Будем генерировать пиксели в соответствии с каким-нибудь порядком (например, слева-направо, сверху-вниз.

Распределение пикселей будем моделировать с помощью LSTM.

Очень медленная тренировка и генерация изображений.



Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

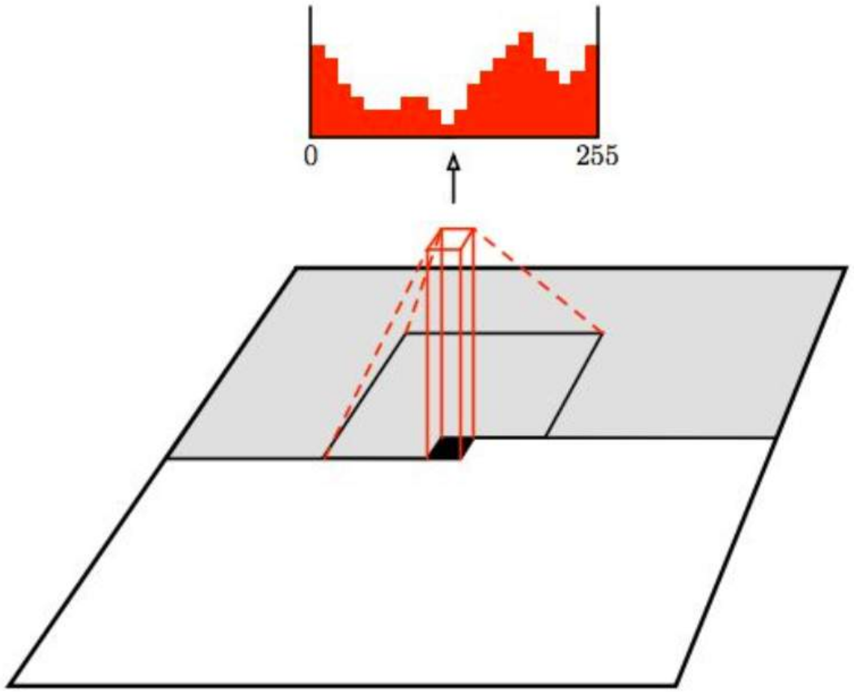
PixelCNN

По-прежнему будем генерировать пиксели в соответствии с каким-нибудь порядком.

Но теперь распределение пикселей будем моделировать с помощью CNN по окну.

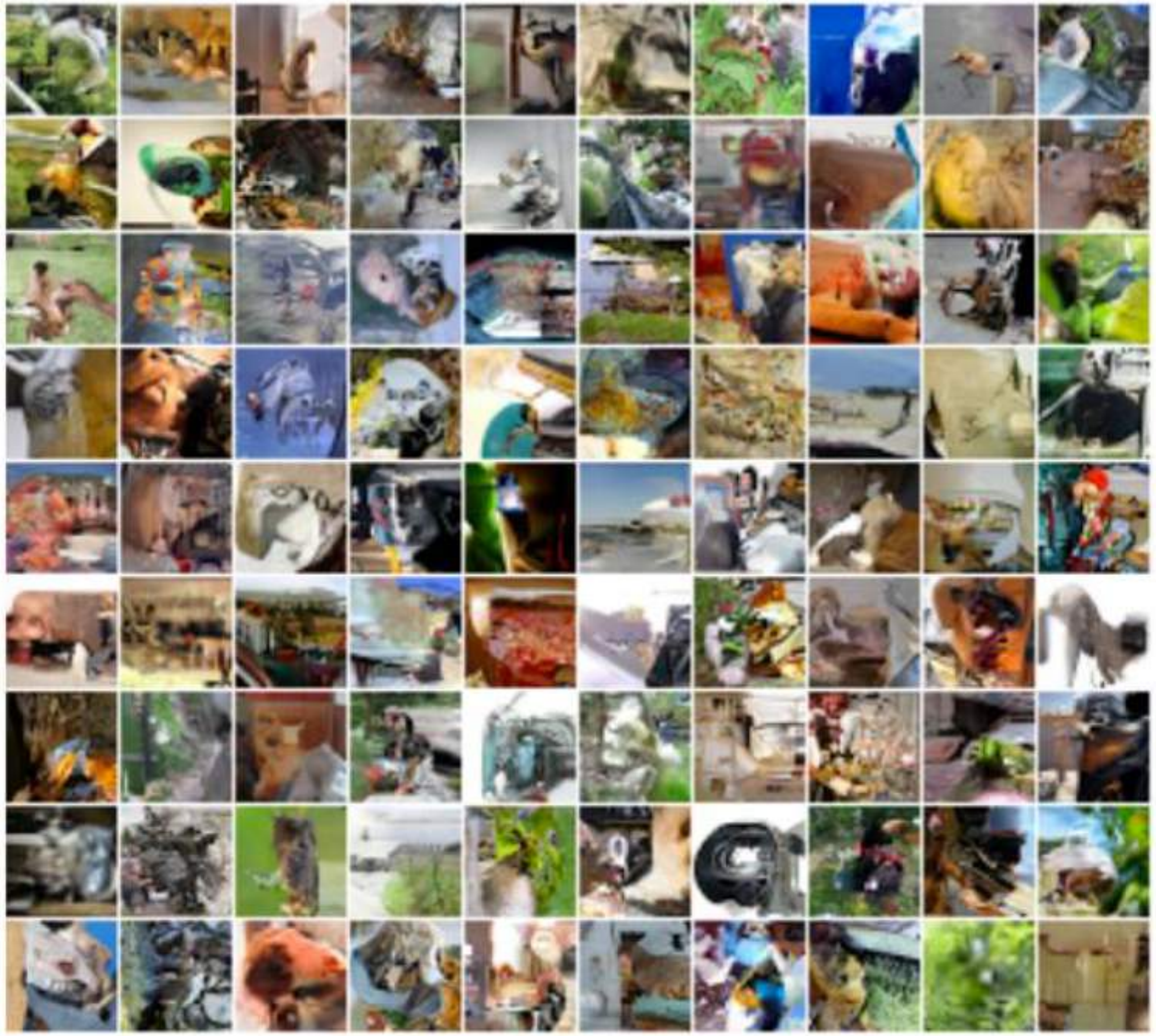
Быстрая тренировка.

Все еще медленная генерация изображений.



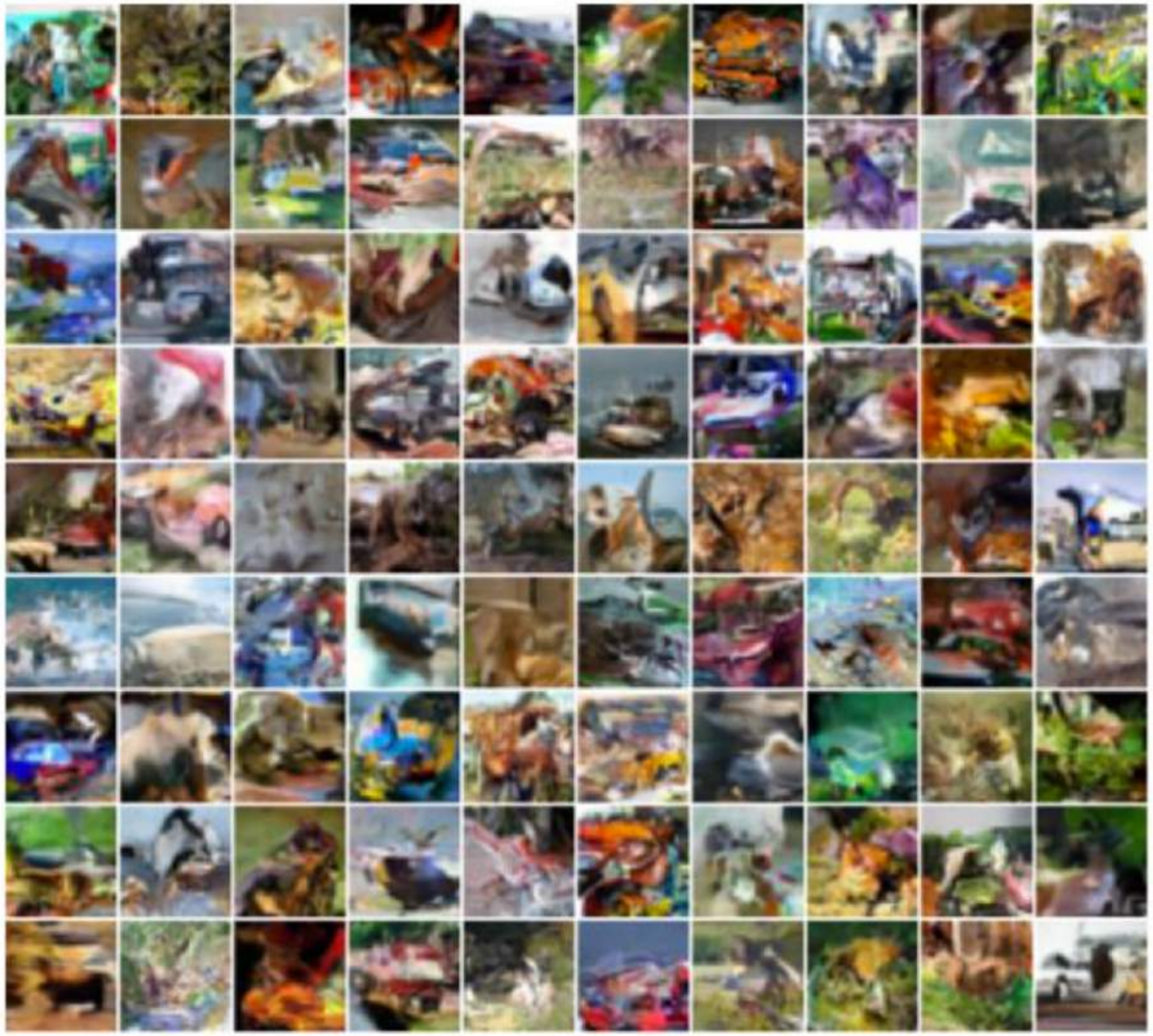
Source: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

PixelRNN and PixelCNN



32x32 ImageNet

PixelRNN and PixelCNN



32x32 CIFAR-10

Useful Materials

cs231 лекция про style transfer, texture synthesis etc.
https://www.youtube.com/watch?v=6wcs6szJWMY&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv

cs231 лекция про генеративные модели
https://www.youtube.com/watch?v=5WoItGTWV54&list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv

Статья про генерацию стульев (Learning to Generate Chairs with Convolutional Neural Networks)
https://www.robots.ox.ac.uk/~vgg/rg/papers/Dosovitskiy_Learning_to_Generate_2015_CVPR_paper.pdf