Lecture 10: Image Synthesis

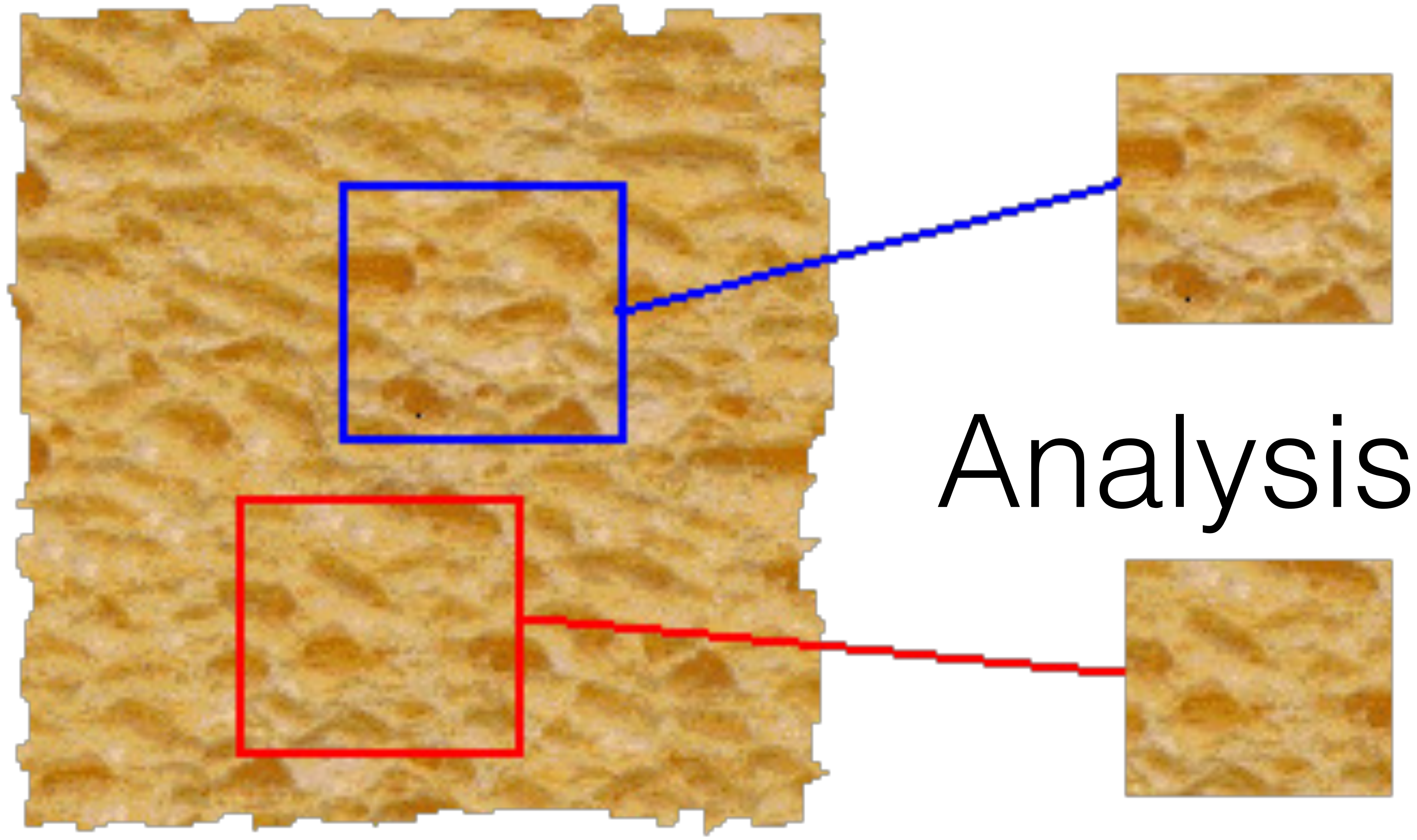
Announcements

- Problem Set 4 is due on Wednesday
  - Need to submit to both Canvas and Gradescope

Today

- Texture synthesis
- Image stylization
- Generative adversarial nets (GANs)
- Conditional GANs

Texture analysis



Analysis

“Same” or “different”

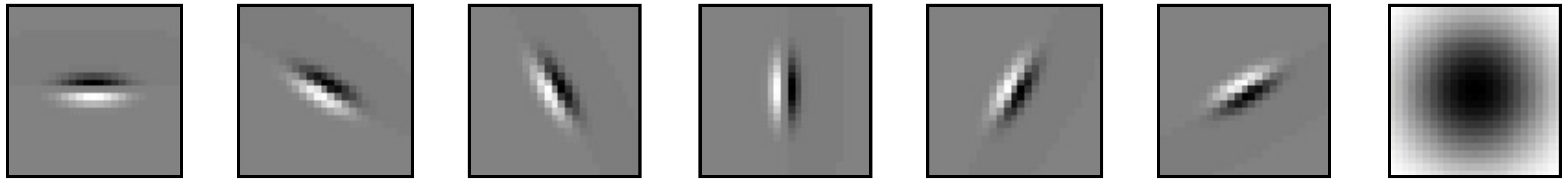
True (infinite) texture

What we’d like: are they made of the same “stuff”. Are these textures similar?

Source: A. Efros

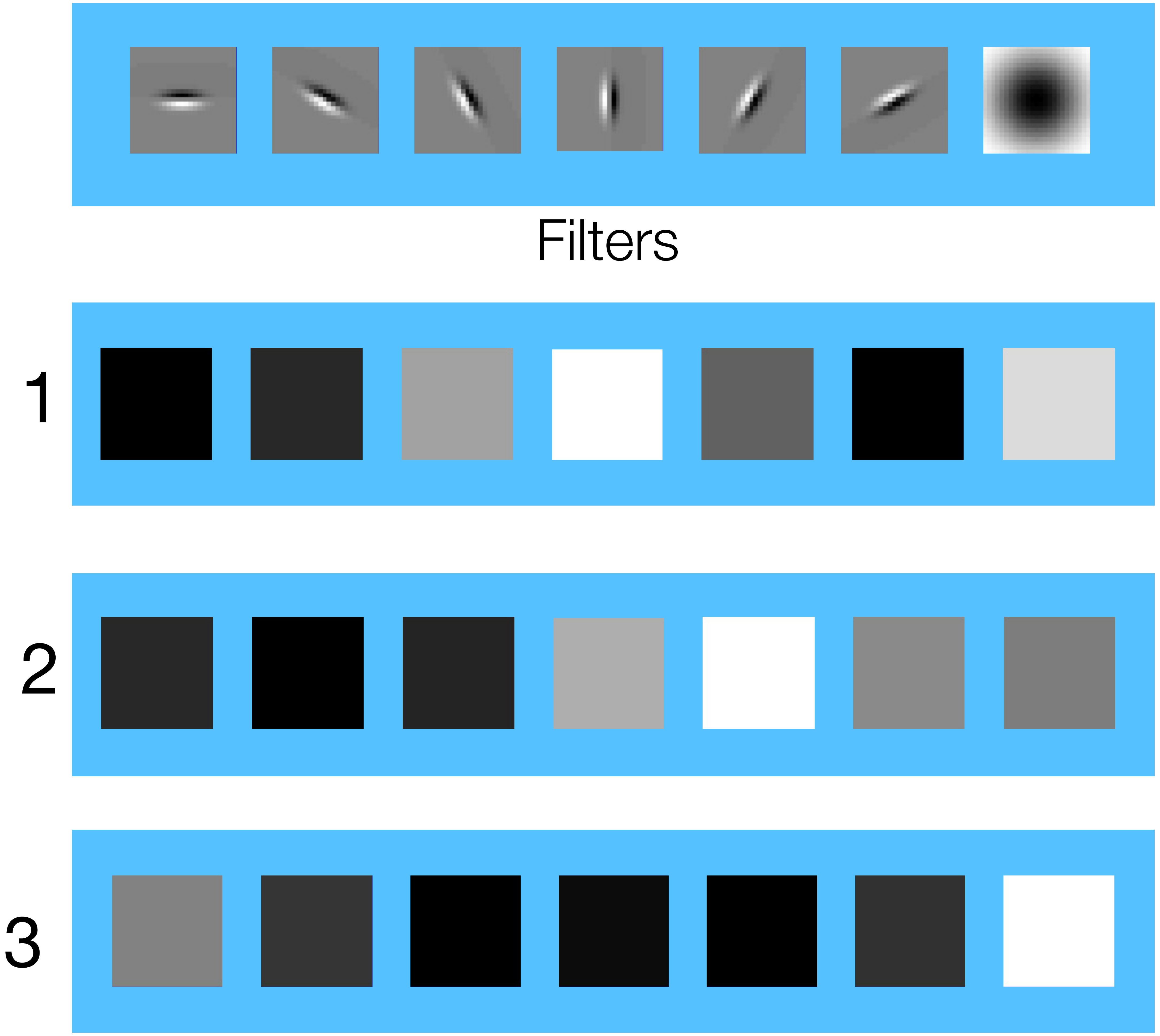
How can we represent texture in natural images?

Idea #1: Record simple statistics (e.g., mean, std.) of absolute filter responses

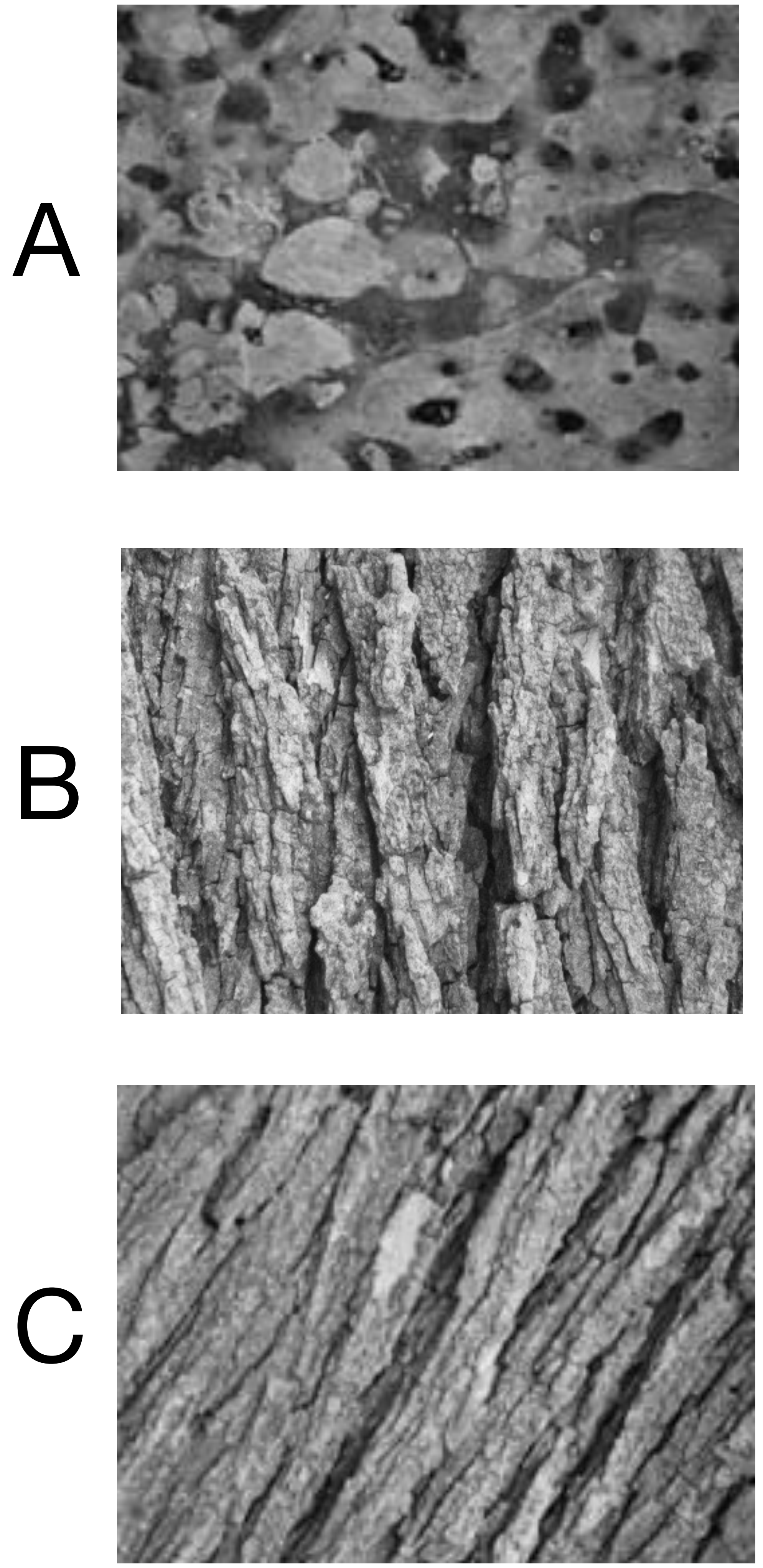


Source: A. Efros

Can you match the texture to the response?



Mean abs. responses



Source: A. Efros

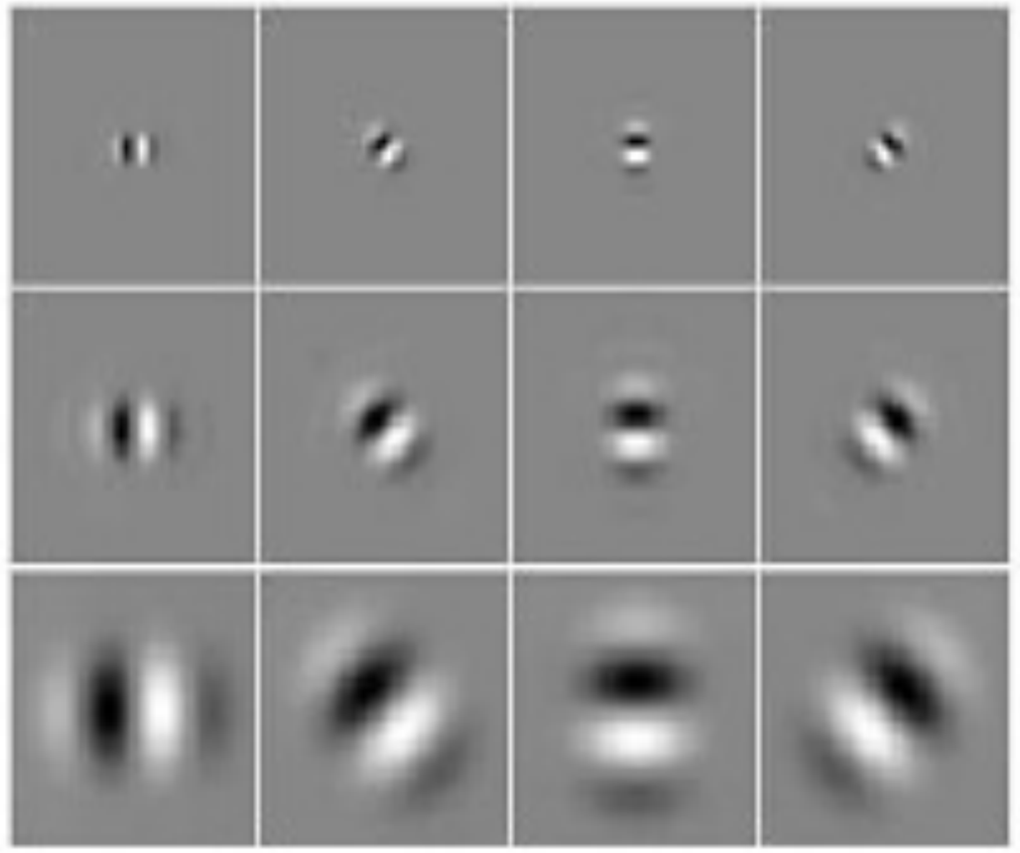
How can we represent texture?

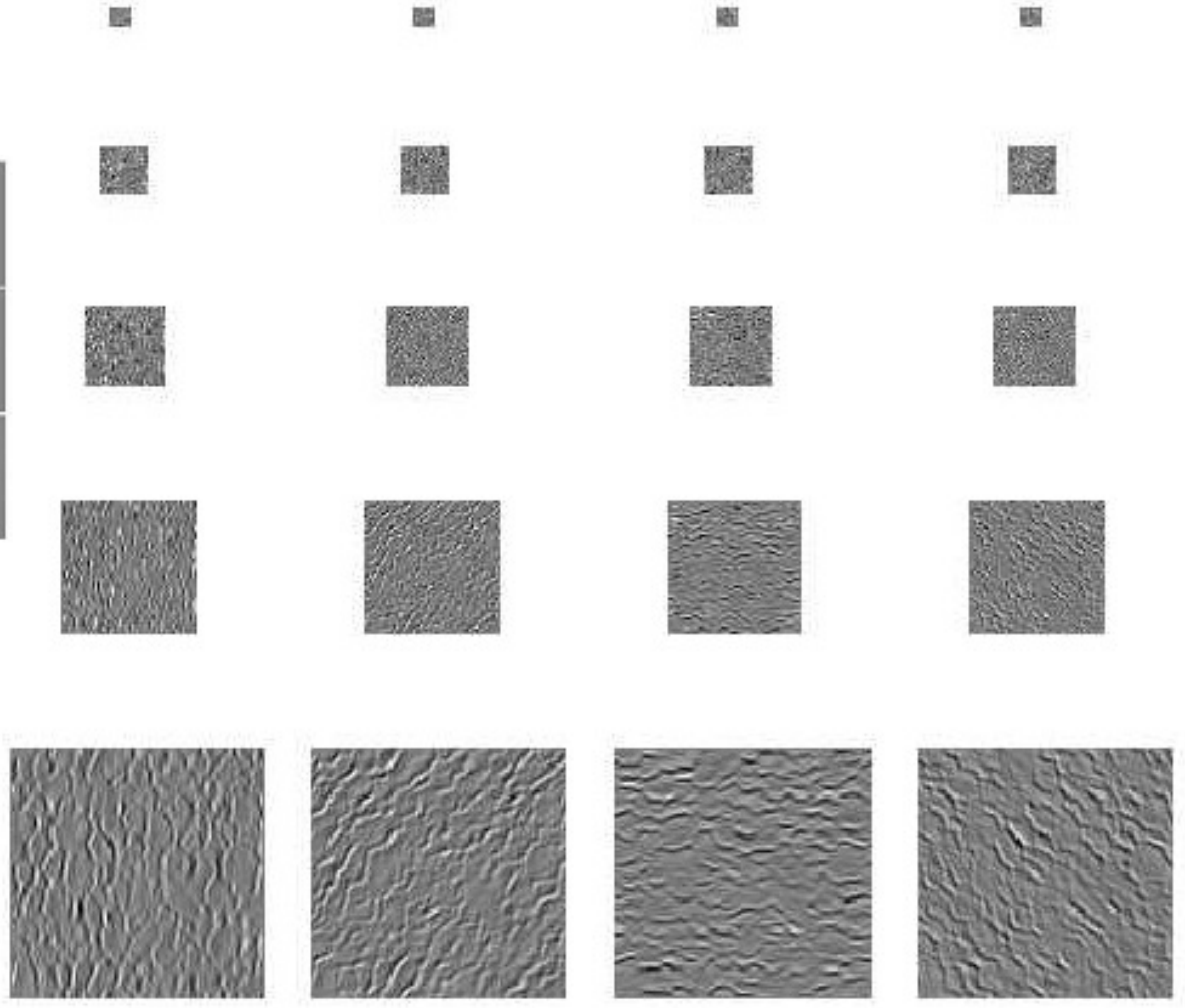
- Generalize this to “orientation histogram”
- Idea #2: Histograms of filter responses
    - One histogram per filter

Source: A. Efros

Pyramid of filter responses

Filter bank

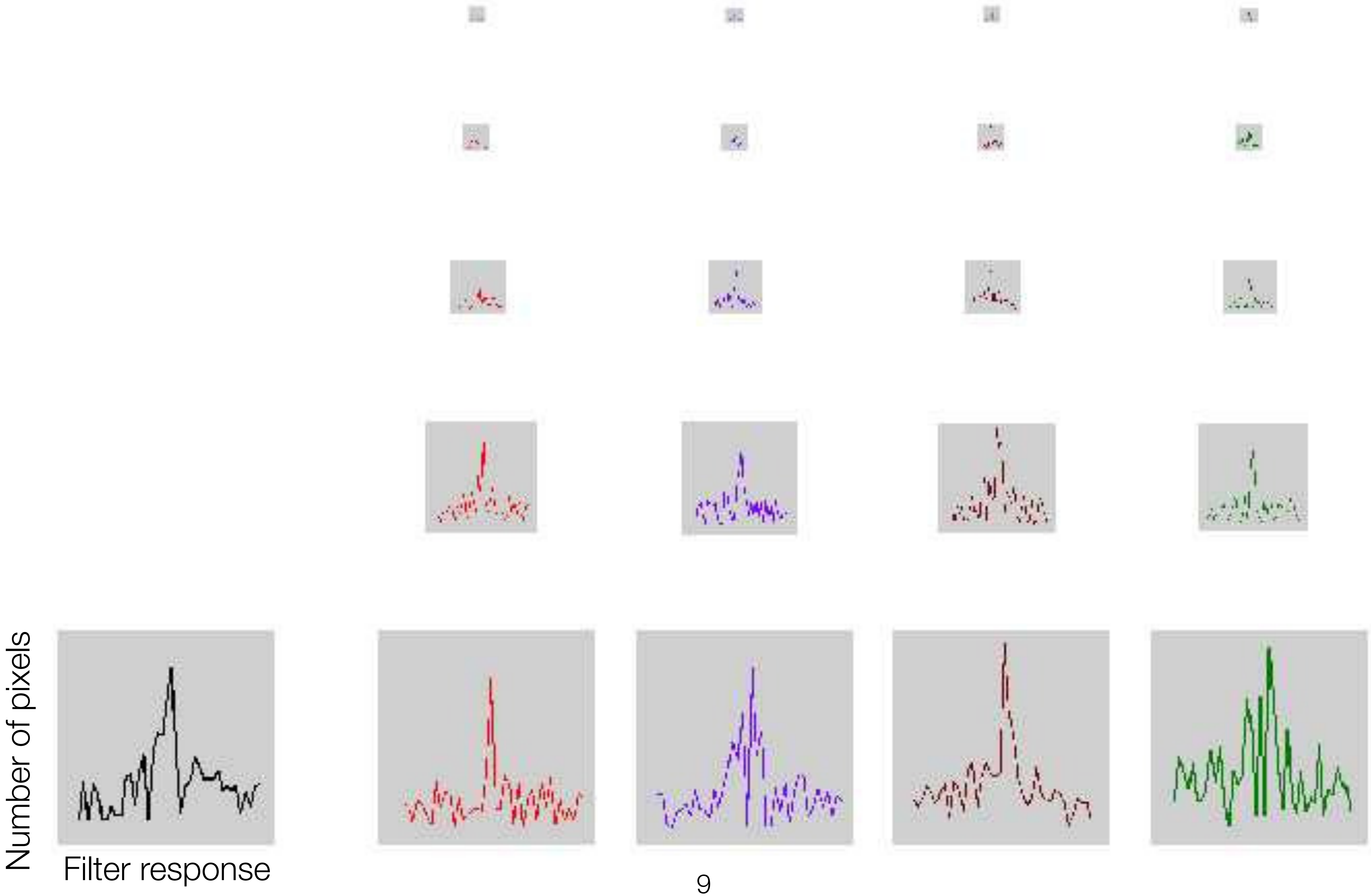




Input image

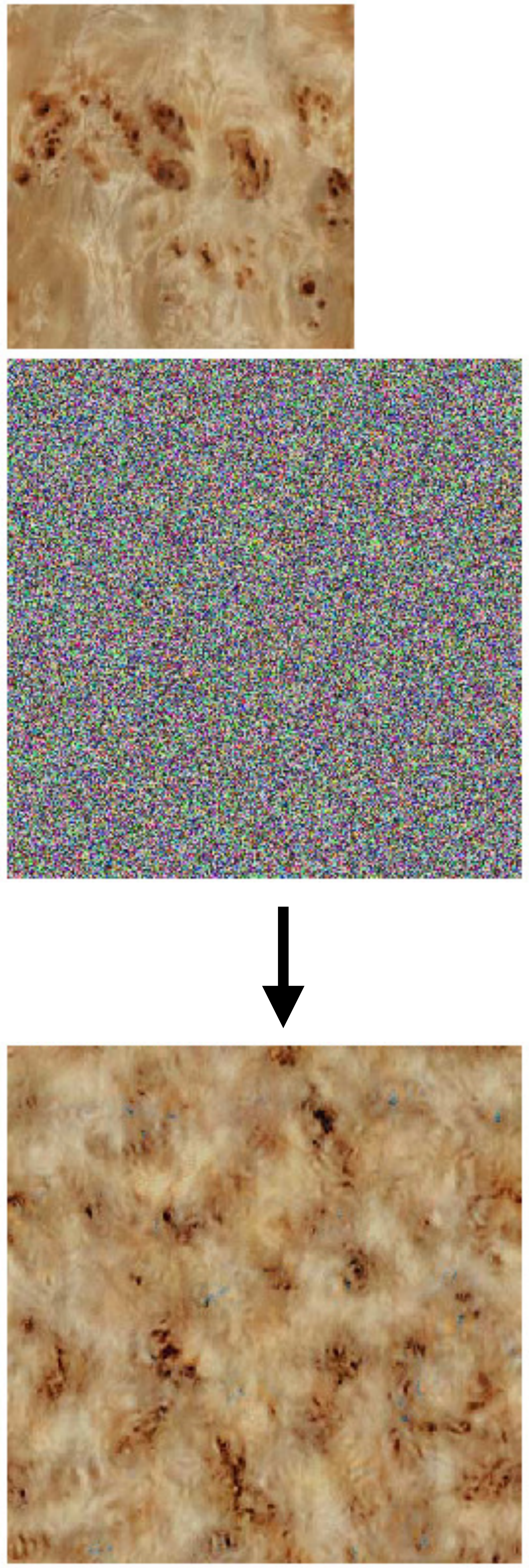
Source: A. Efros

Filter response histograms



Source: A. Efros

Texture synthesis

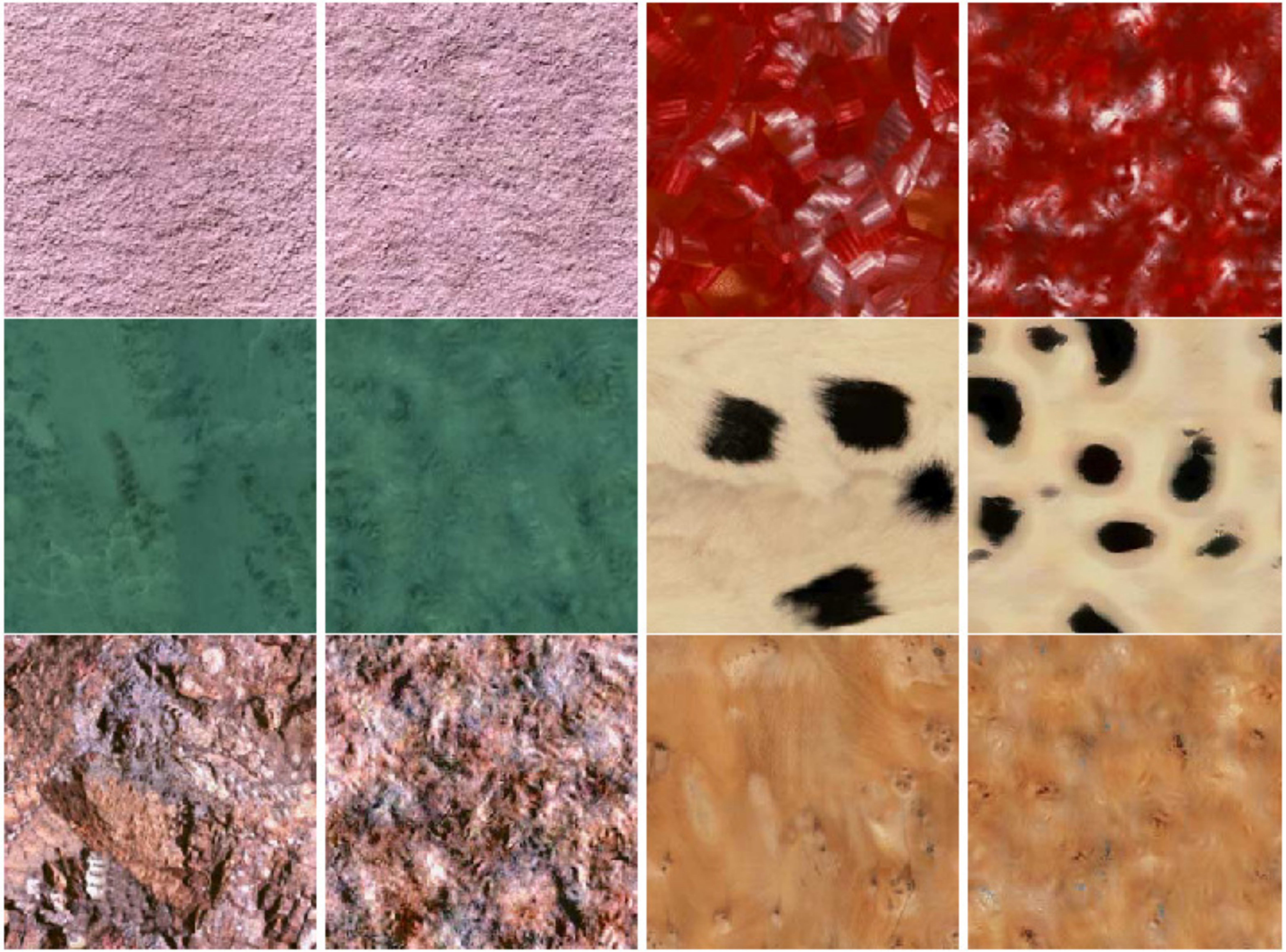


Start with a noise image as output.

Iterative algorithm [Heeger & Bergen, 95]:

- Match pixel histogram of output image to input
- Decompose input/output images using a Steerable Pyramid
- Match histograms of input and output pyramids
- Reconstruct image and repeat

Source: A. Efros



Source: A. Efros

Source: A. Efros

Failure cases

Figure 7: (Left pair) Inhomogeneous input texture produces blotchy synthetic texture. (Right pair) Homogenous input.

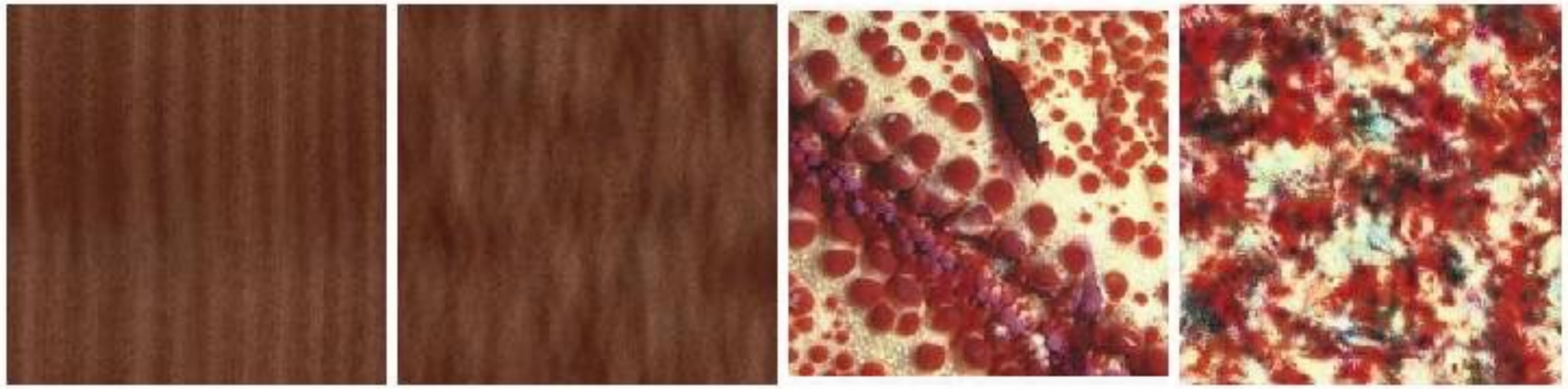


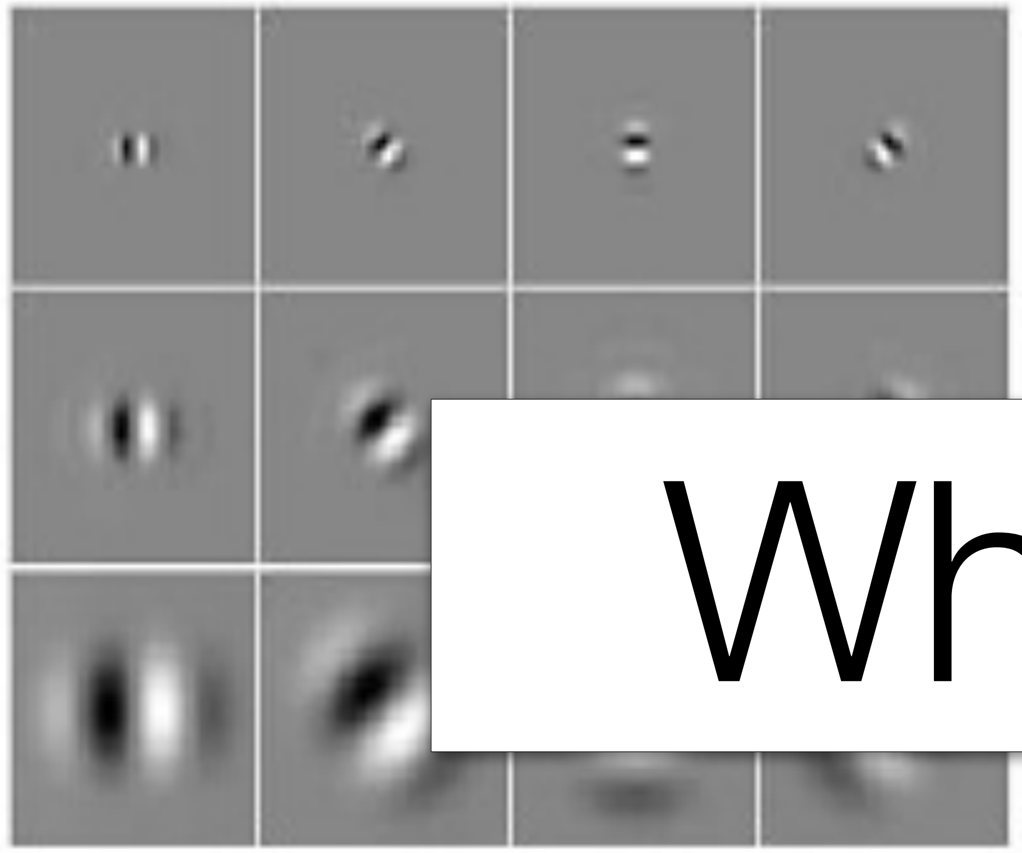
Figure 8: Examples of failures: wood grain and red coral.

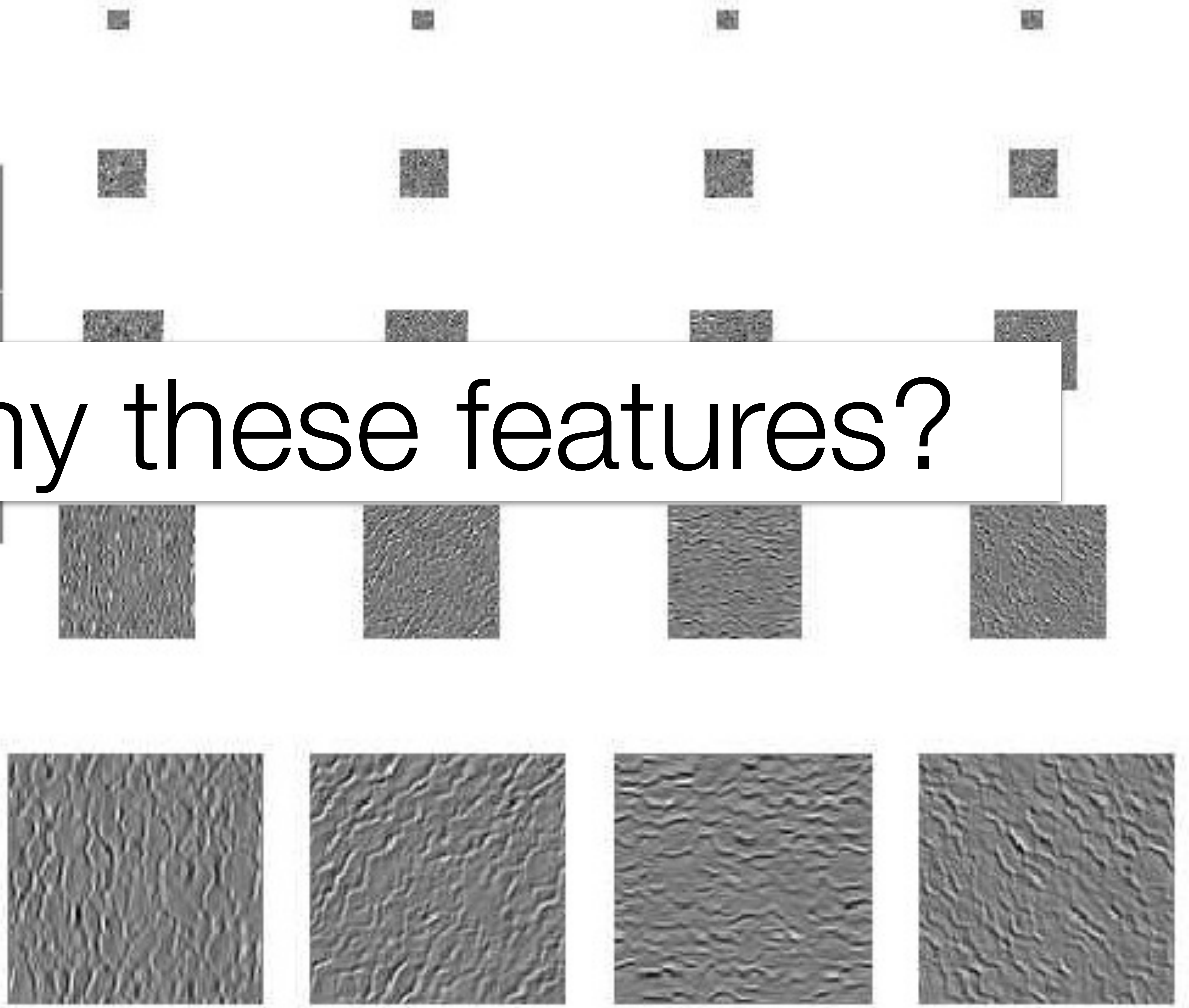
Figure 9: More failures: hay and marble.

Source: A. Efros

Pyramid of filter responses

Filter bank



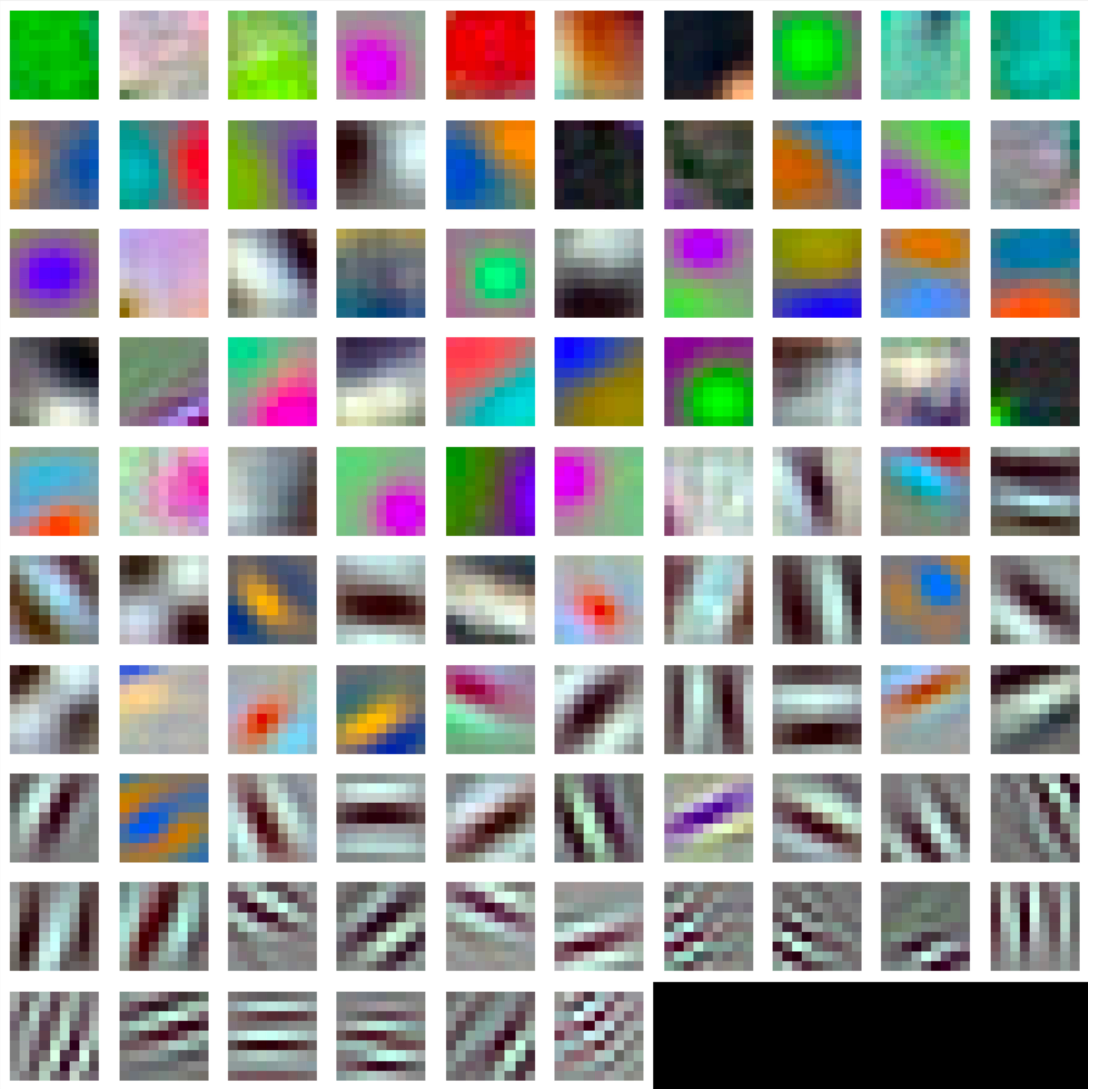


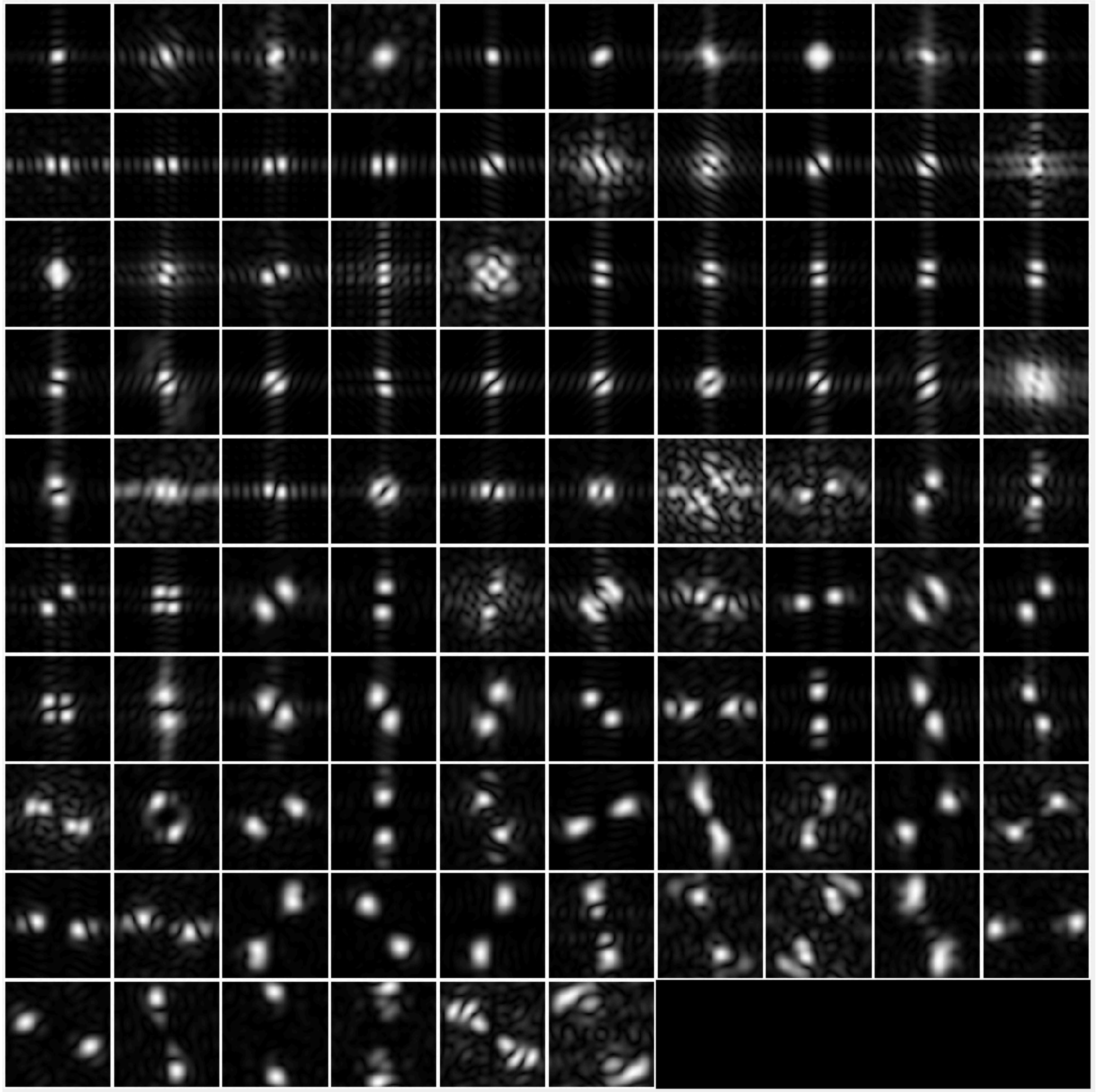
Why these features?

Input image

Source: A. Efros

Recall: neural net feature visualization

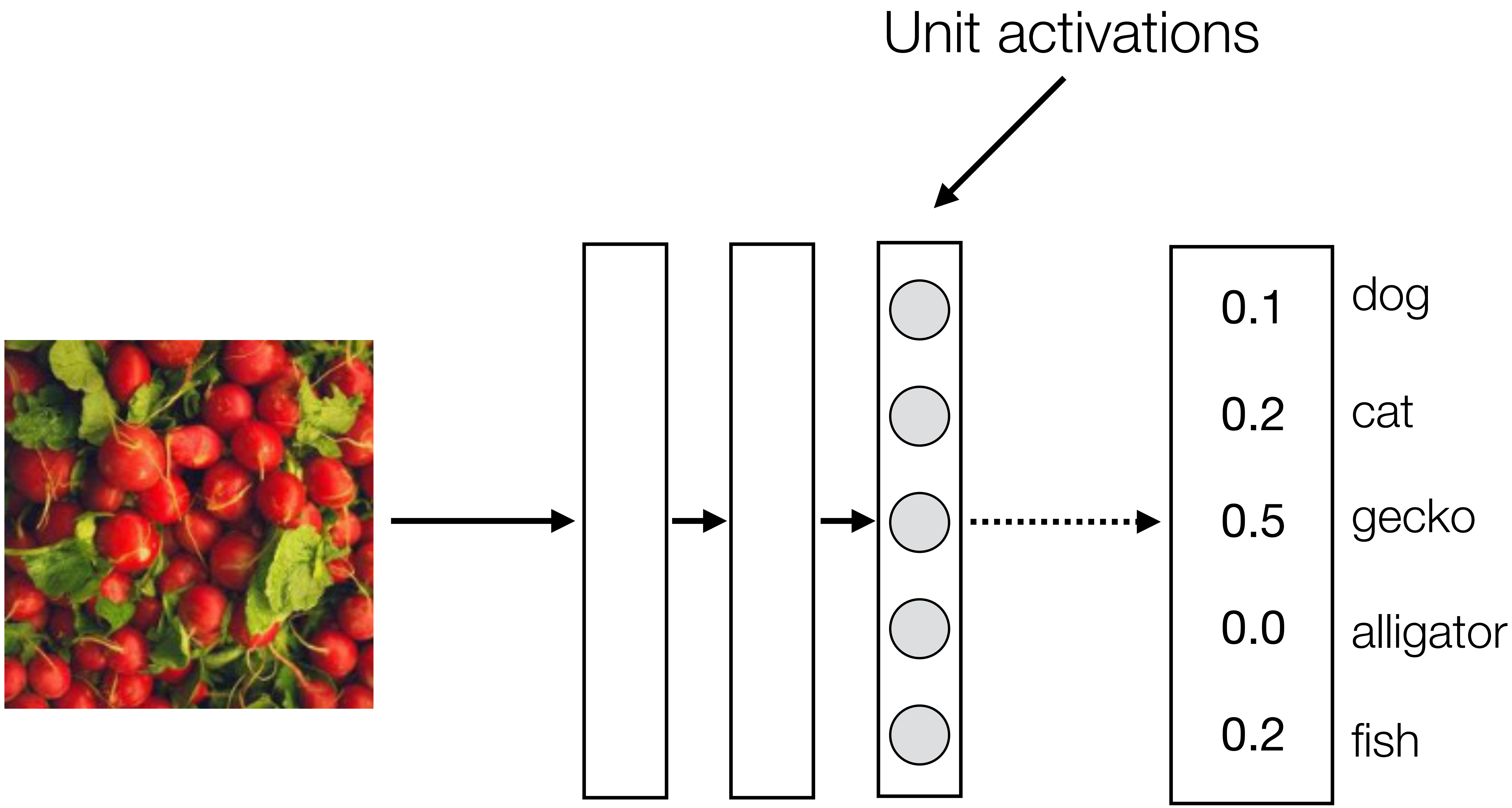




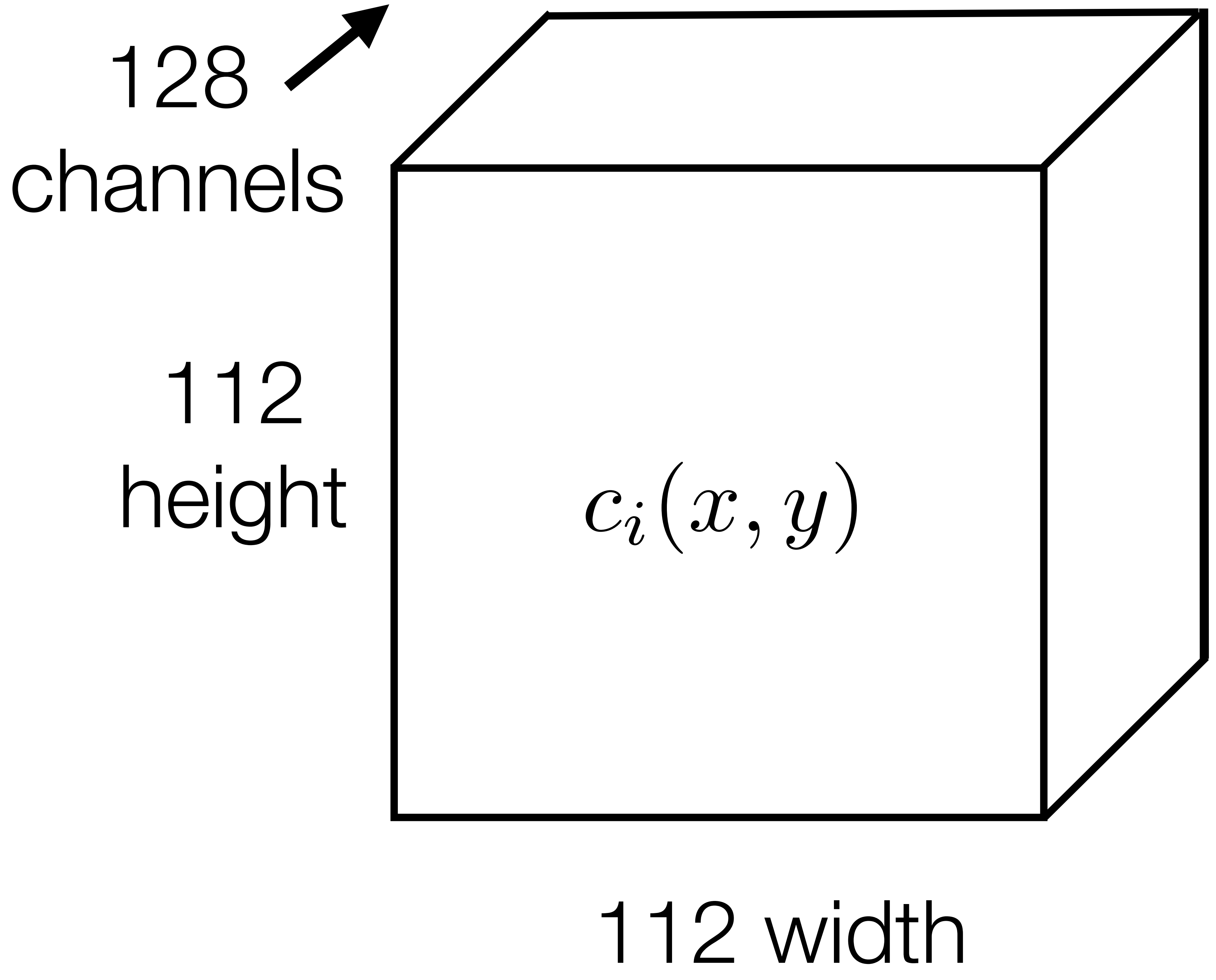
96 Units in conv1

Source: Isola, Torralba, Freeman

Extracting features from a trained network

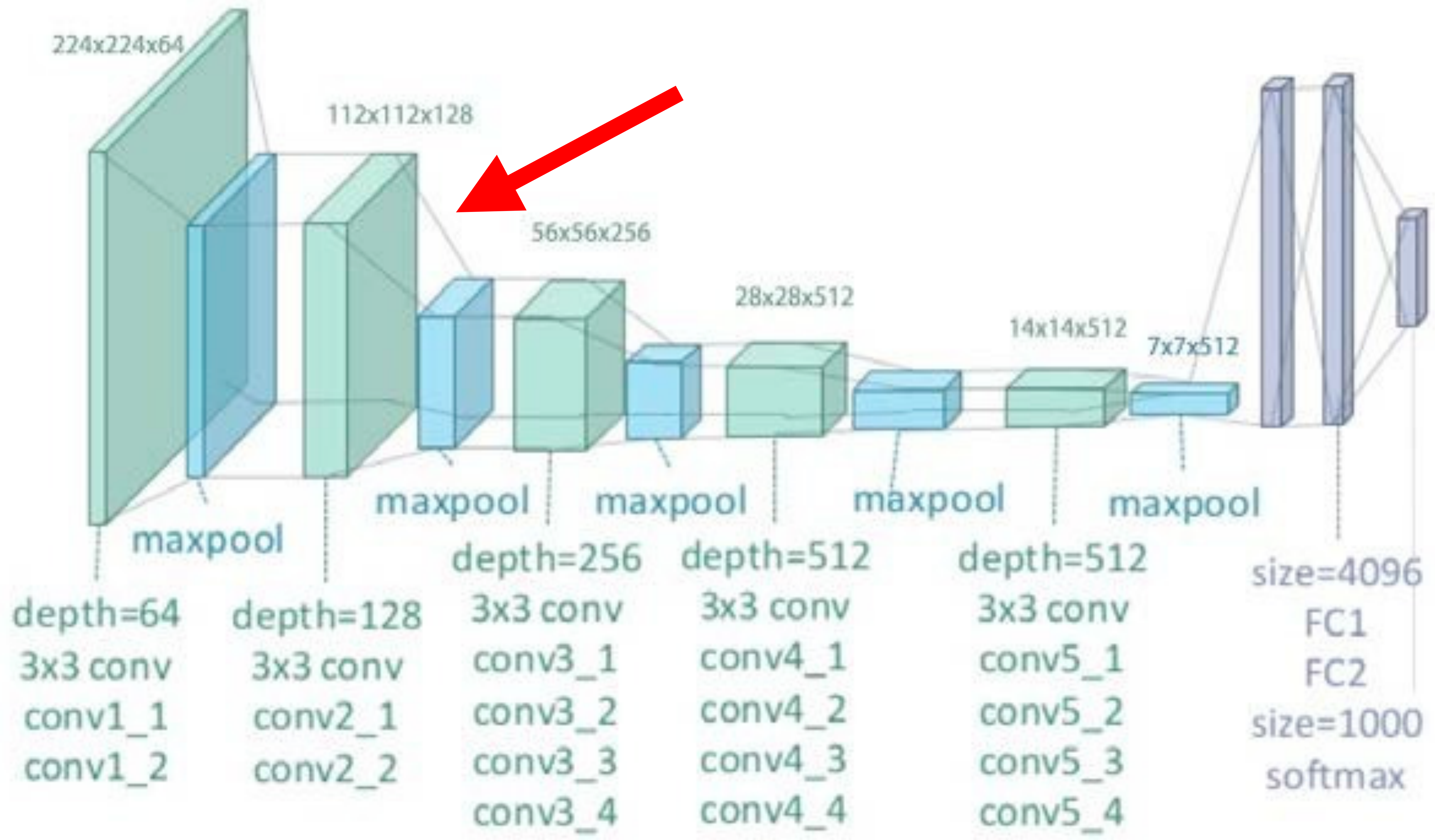


Extracting neural net features

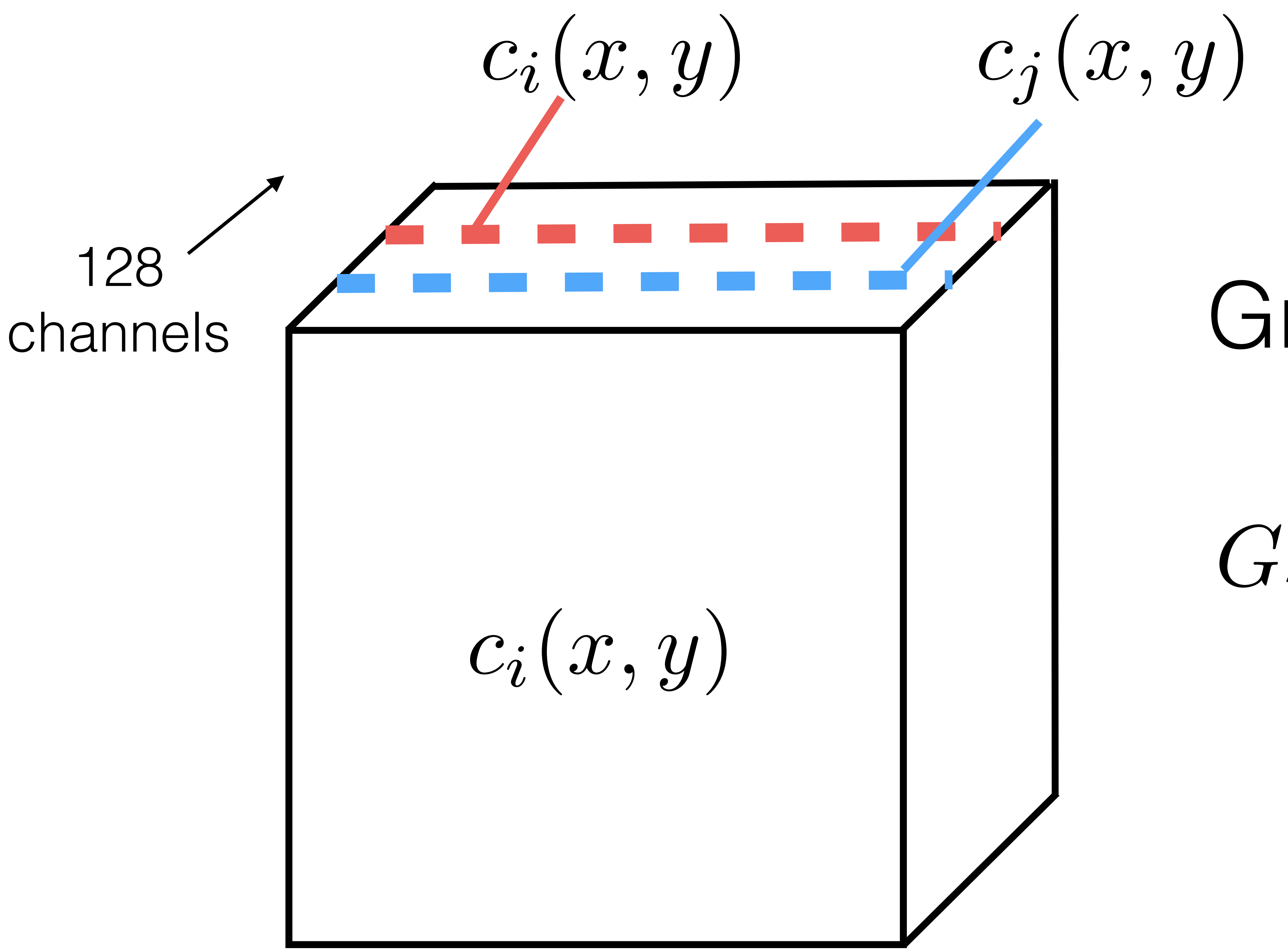


conv2 feature activations

(112 x 112) x 128 for conv2 of VGG19



Capturing feature correlations



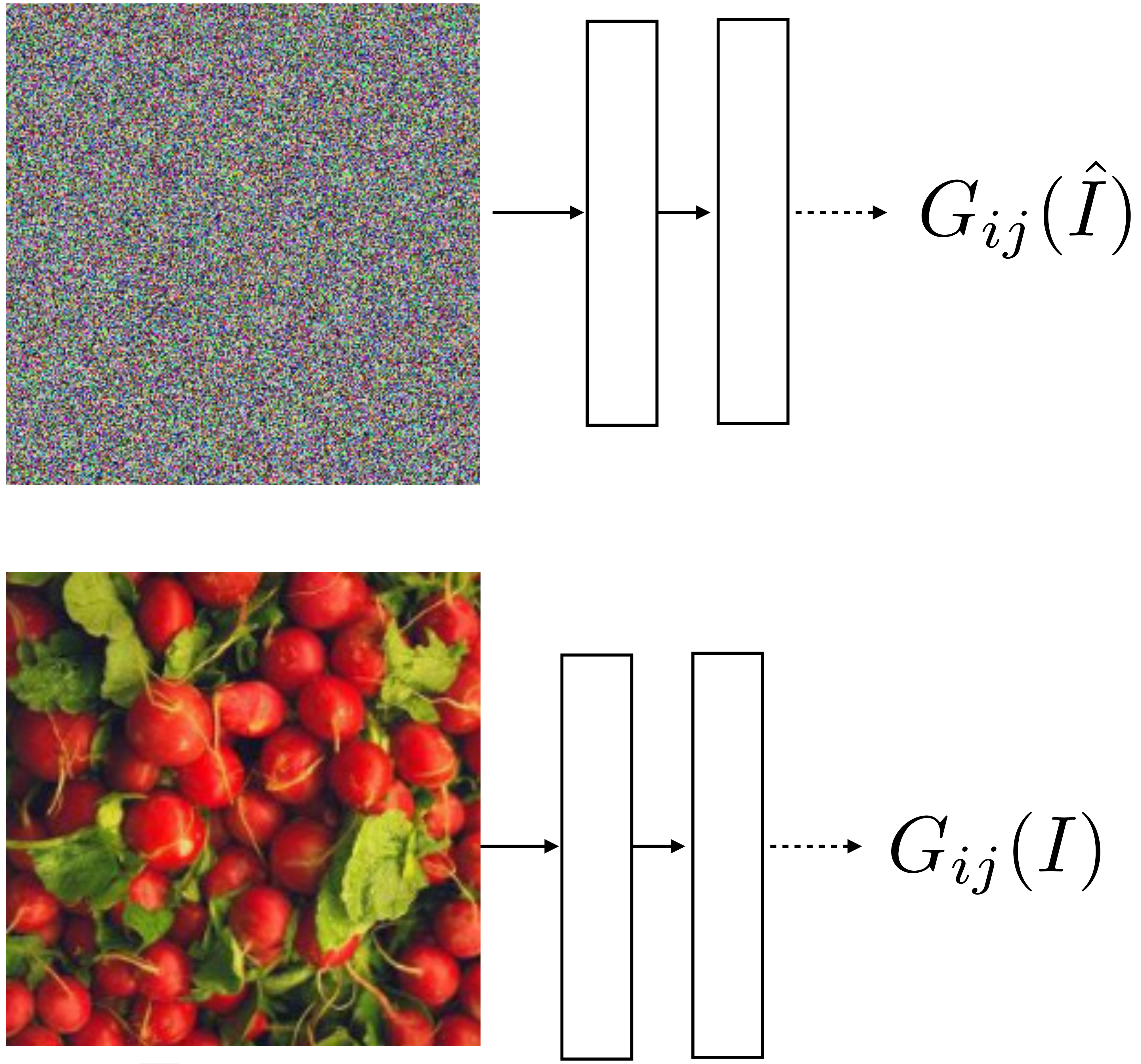
Gram (≈ covariance) matrix:

$$G_{ij} = \sum_{x=1}^{w} \sum_{y=1}^{h} c_i(x, y) c_j(x, y)$$

[Gatys et al. 2016]

Idea: correlations between unit activations convey texture.
Discard global spatial information.

Matching image statistics



Target

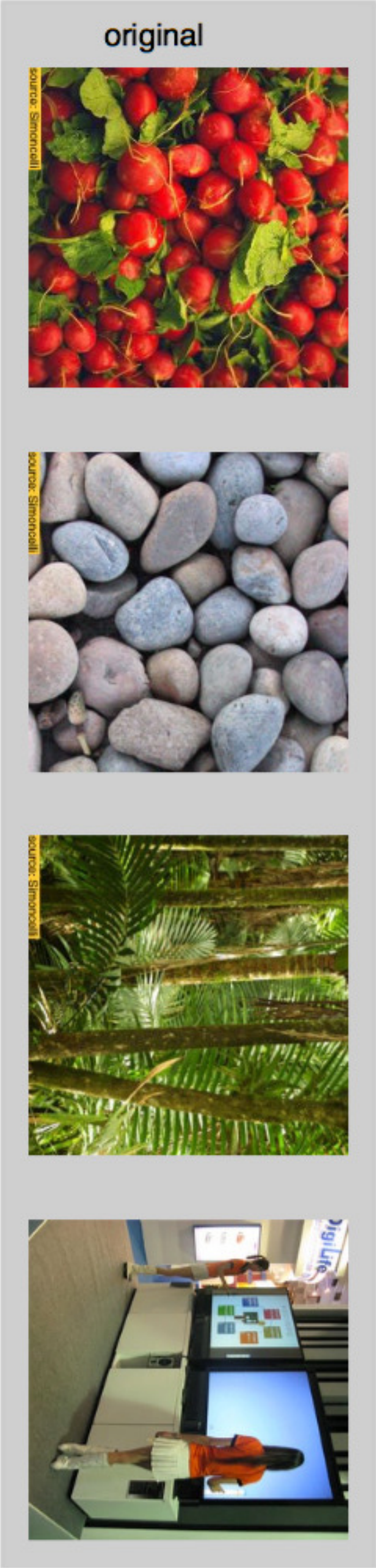
Find $\hat{I}$ by minimizing:

$$\sum_{i=1}^{128}\sum_{j=1}^{128}(G_{ij}(I) - G_{ij}(\hat{I}))^2$$

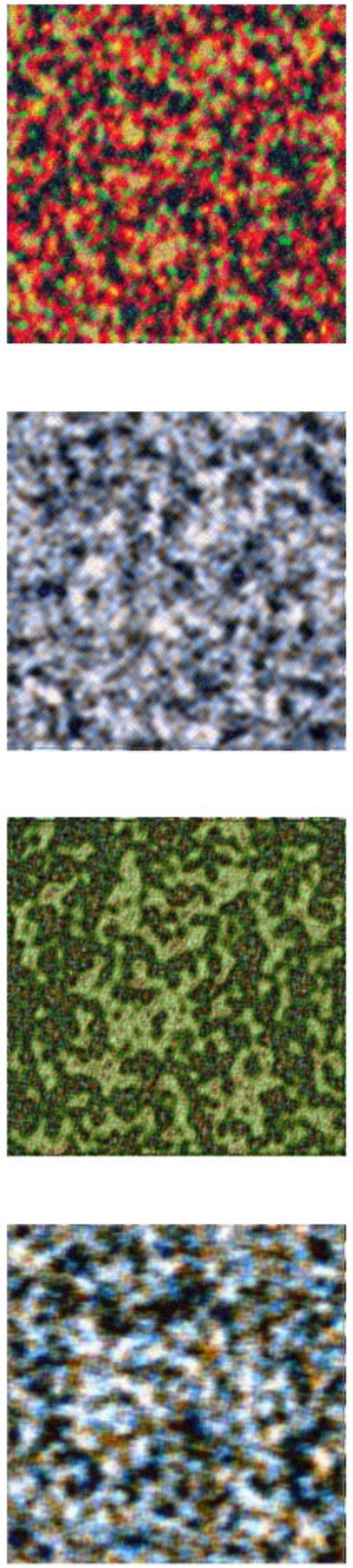
Implementation details:

- Minimize with gradient descent
- How do we compute gradients? Backprop!
- Use many layers of network.

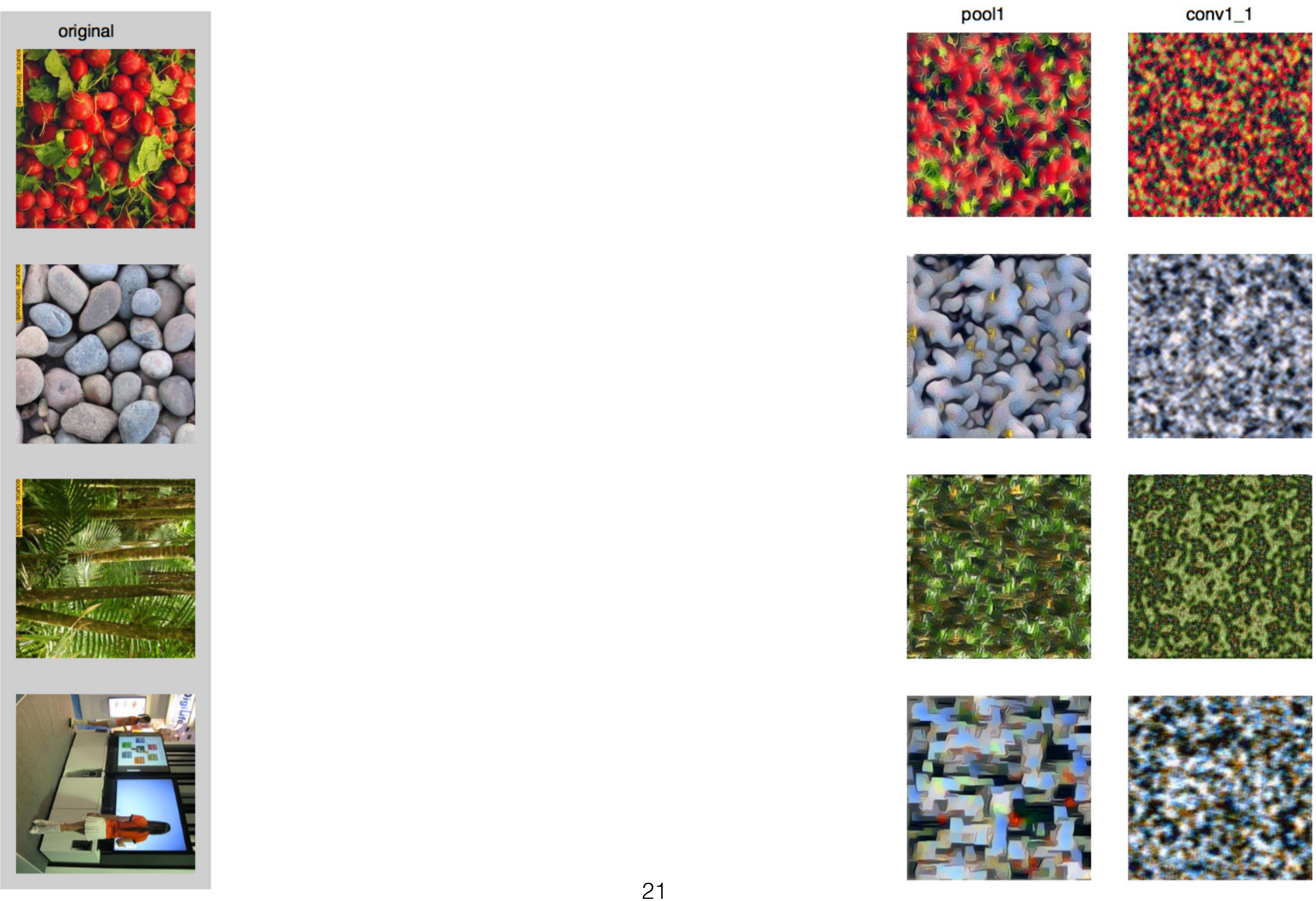
High → Low



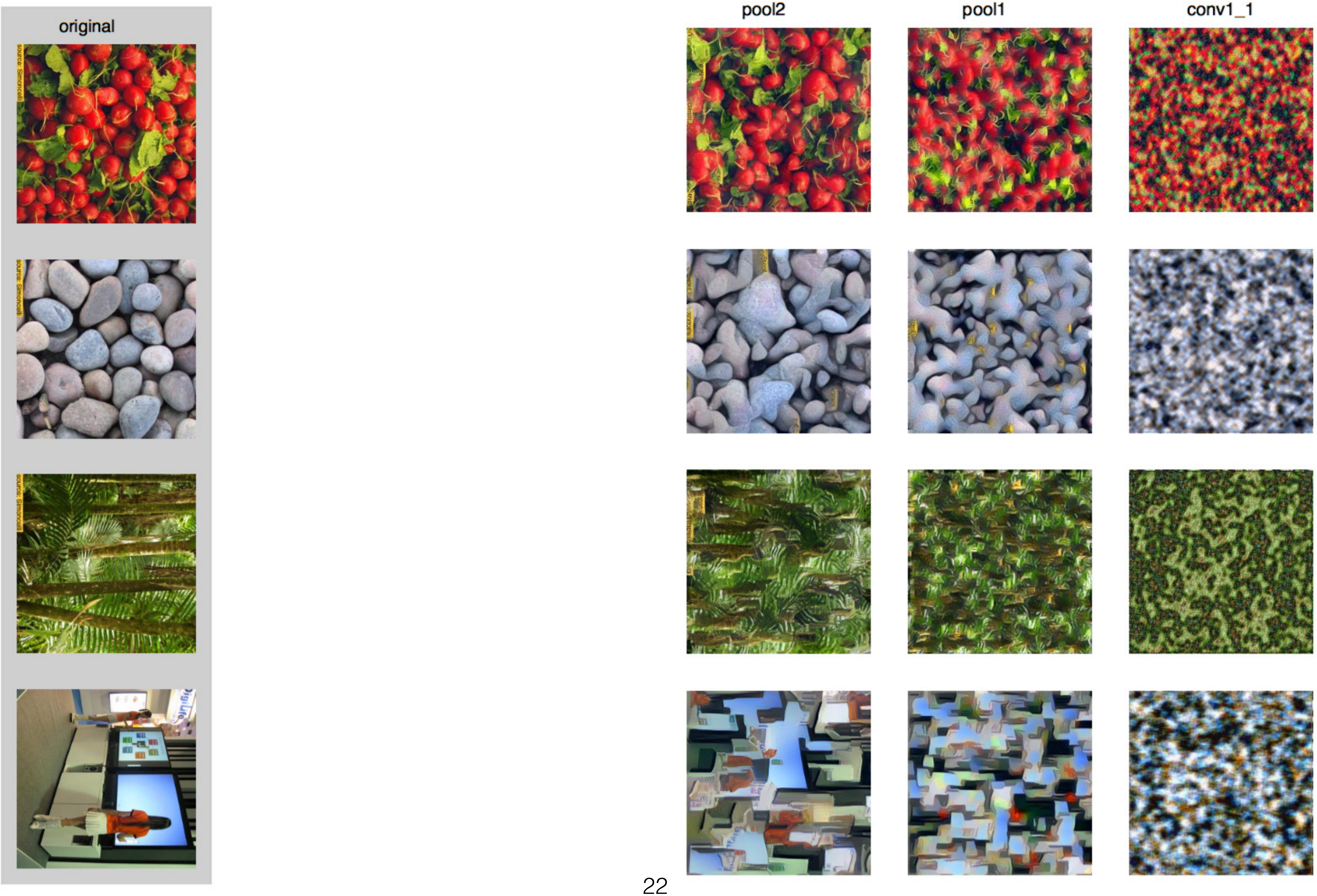
conv1_1

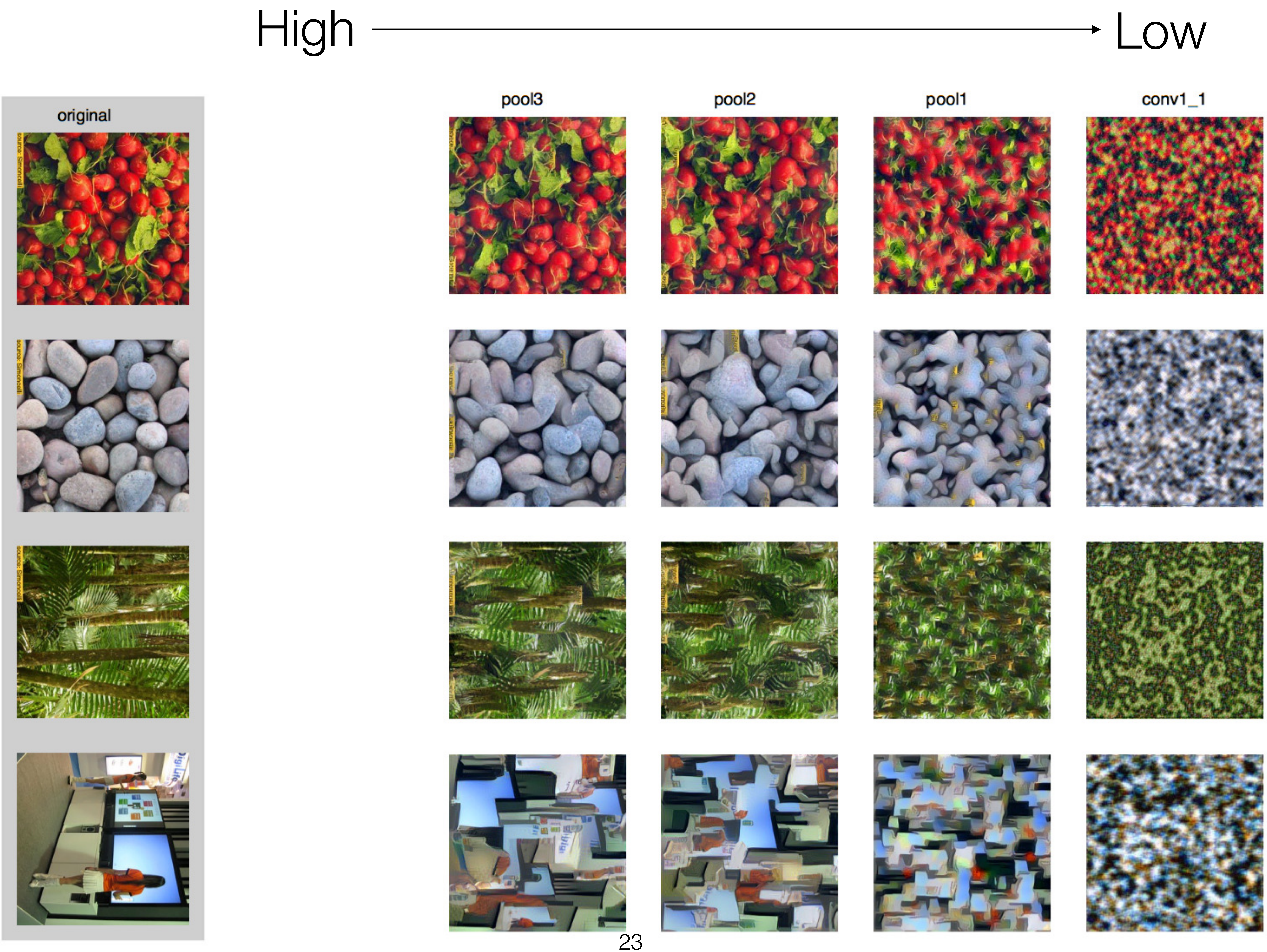


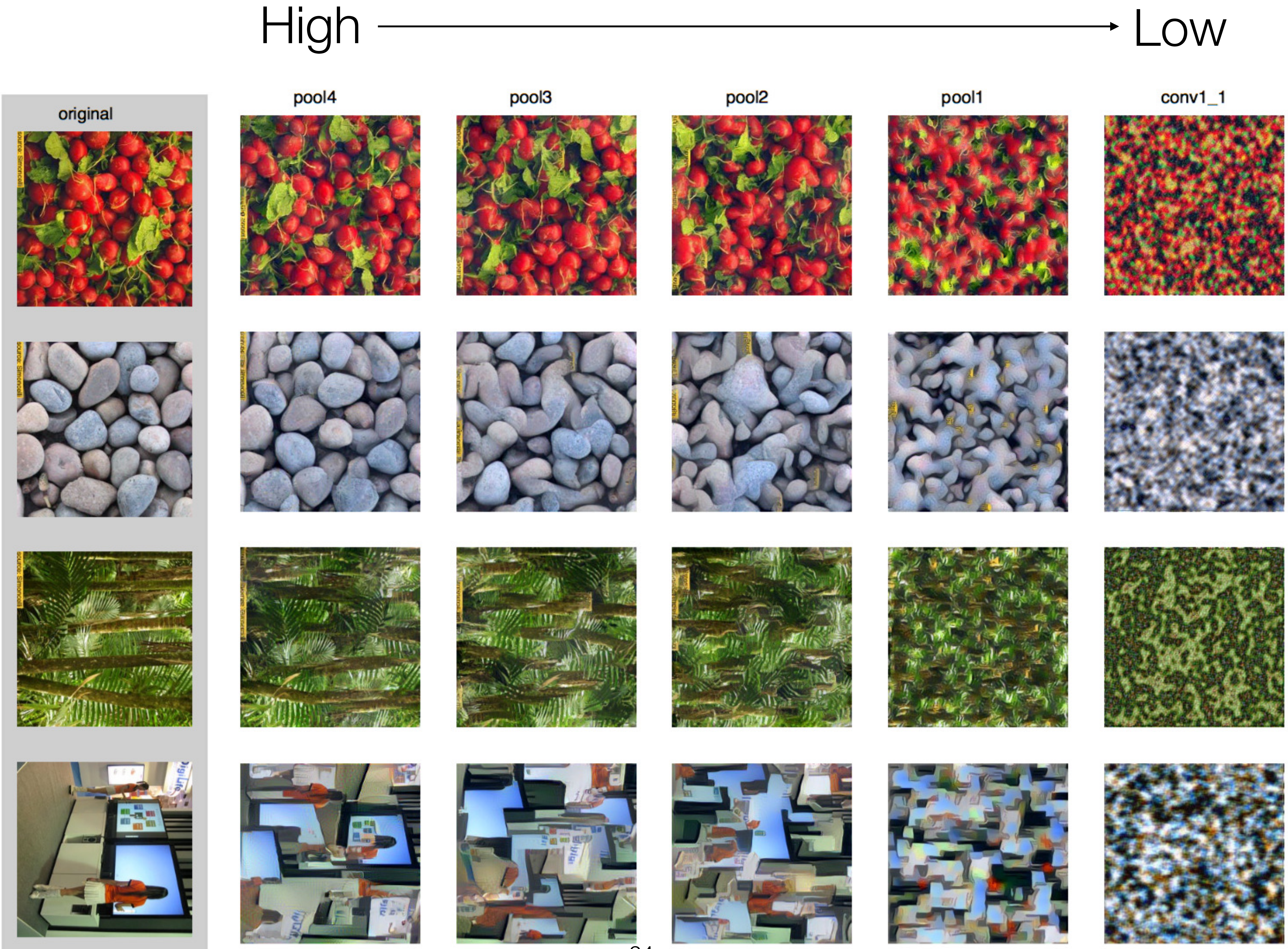
High → Low

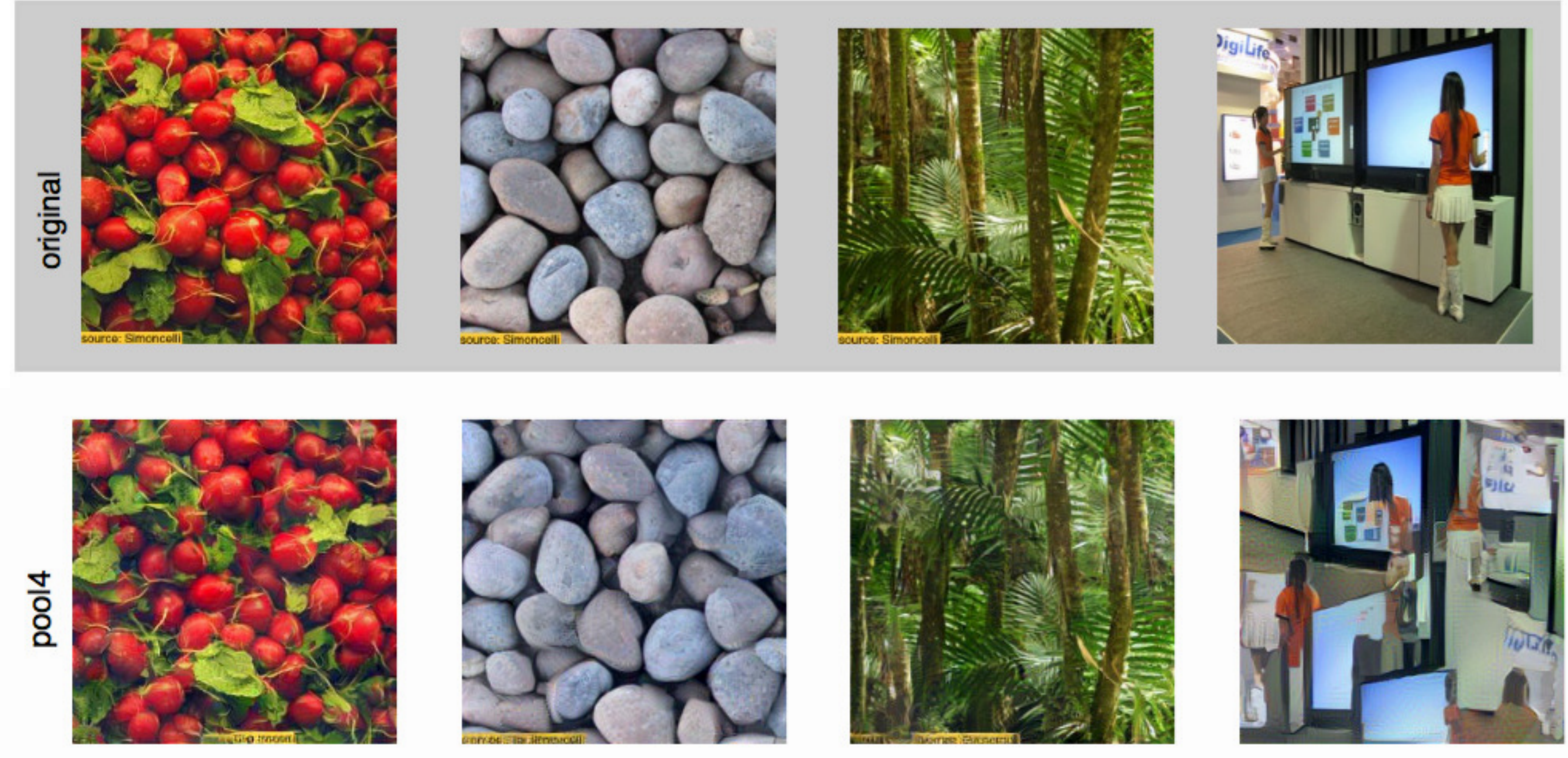


High → Low




High
Low
original
pool3
pool2
pool1
conv1_1


High
Low
original
pool4
pool3
pool2
pool1
conv1_1

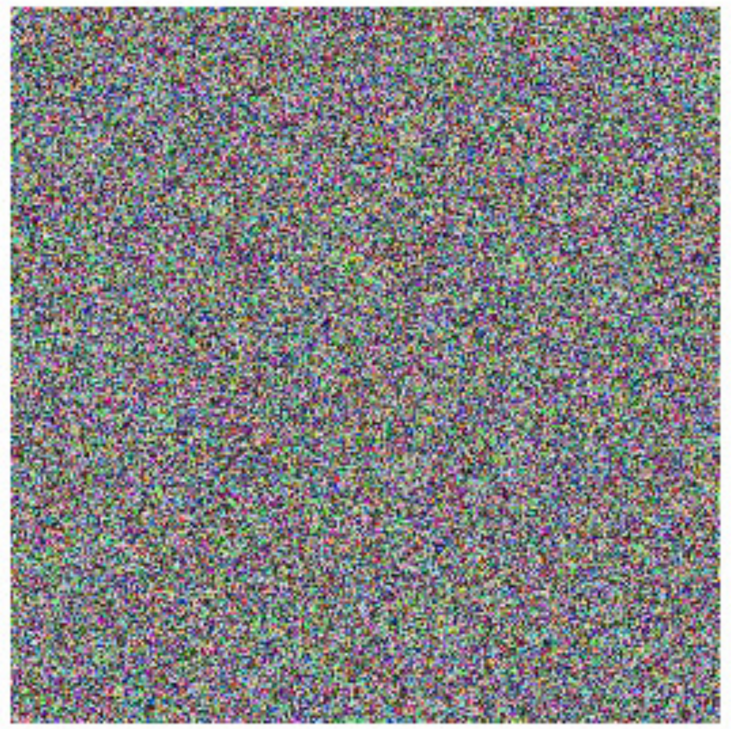

original
pool4
source: Simoncelli
source: Simoncelli
source: Simoncelli

Texture captures artistic style

Can we transfer the style of a painting to a photo?

[Gatys et al. 2016]

Match the **style** of the painting.



Synthesized image

$G_{ij}(\hat{I})$

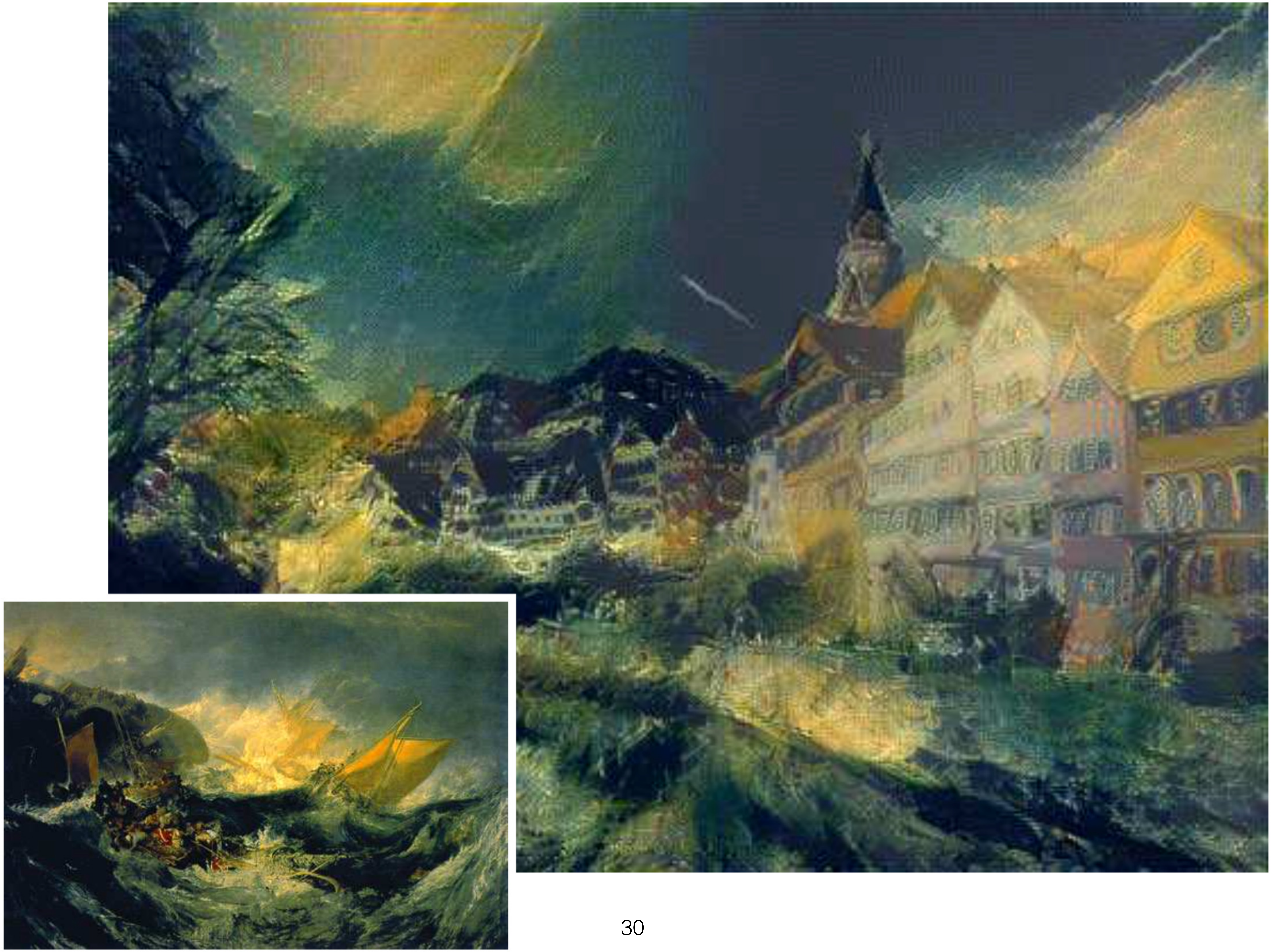
$$\sum_{i=1}^{128}\sum_{j=1}^{128}(G_{ij}(\hat{I}) - G_{ij}(I))^2$$

Perceptual loss:
usually distance in feature space

… and the **content** of the photo.

$$\sum_{i}\sum_{x,y}(c_i(x,y) - \hat{c}_i(x,y))^2$$

$c_i(x,y)$ $\hat{c}_i(x,y)$

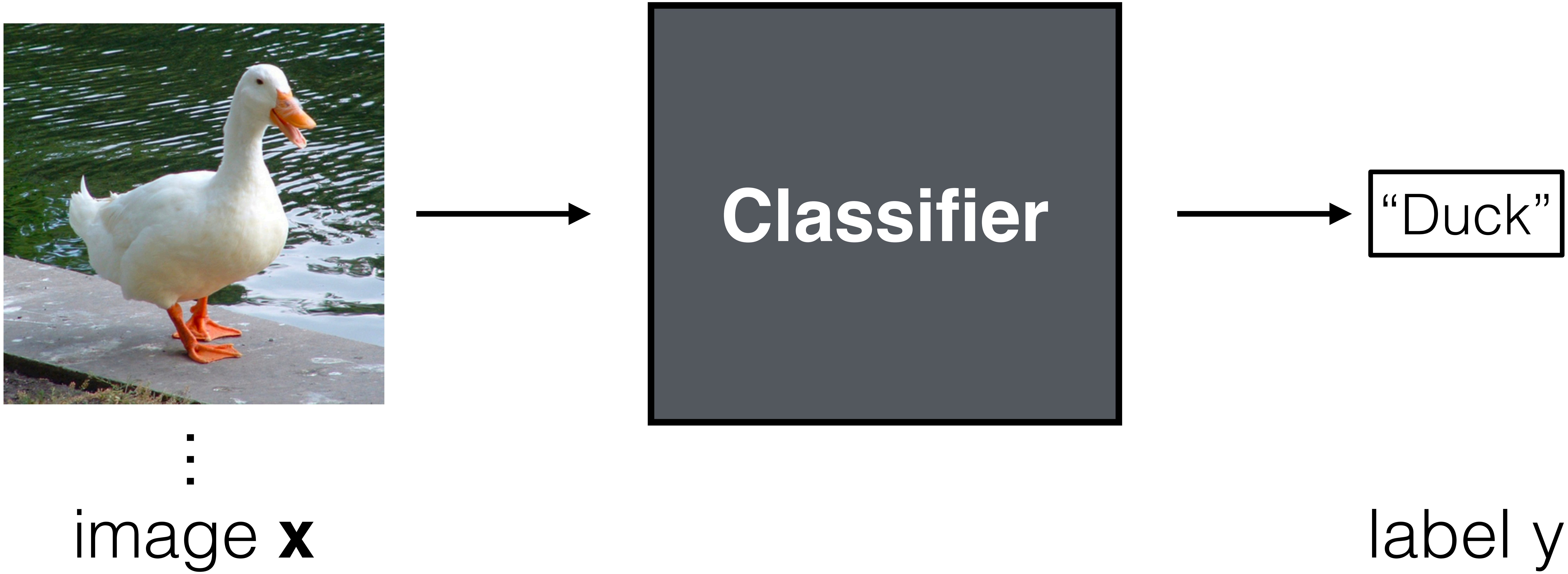



London during the day.

New York at night.

Neural networks that generate images

Image classification

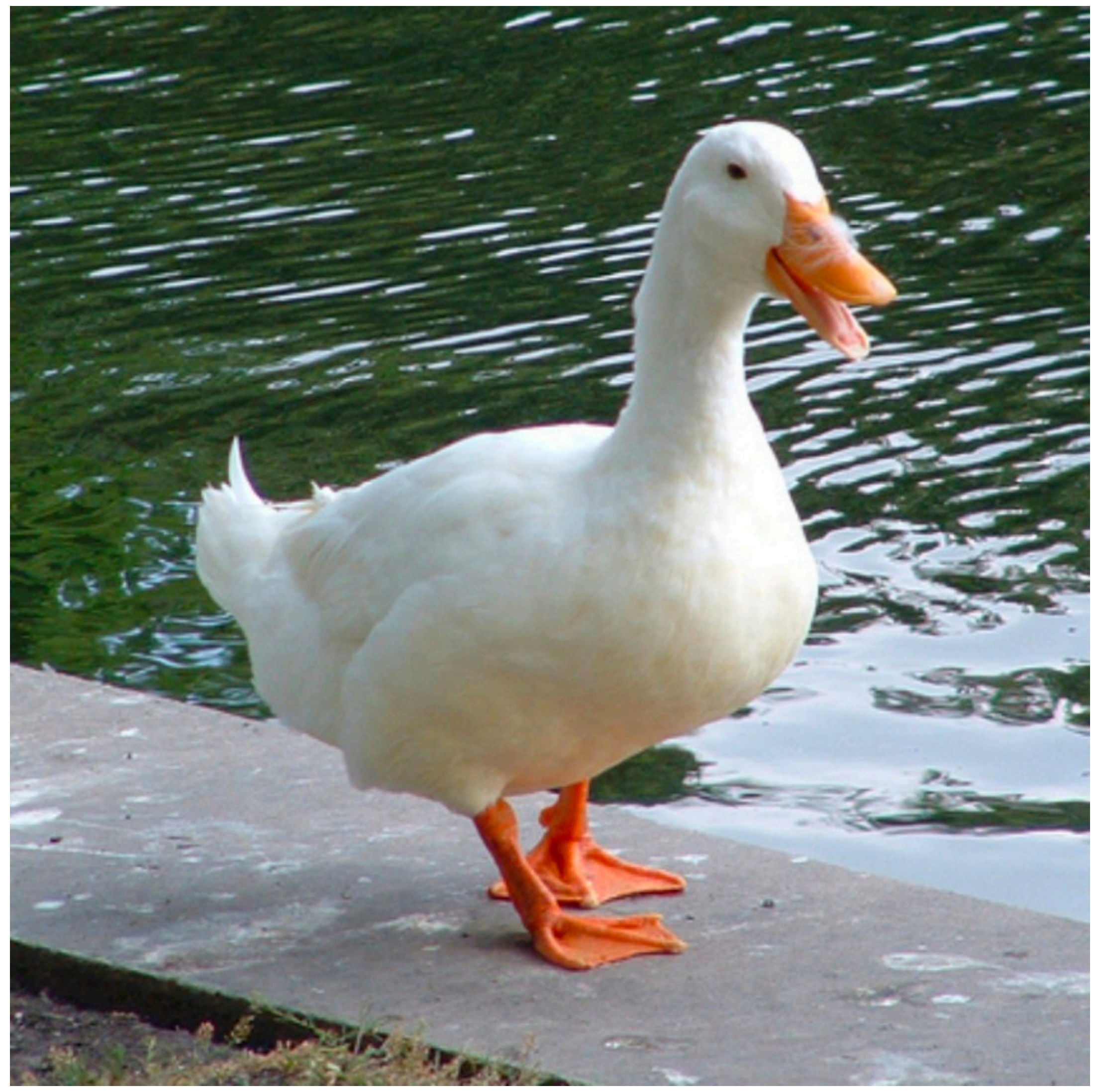


Source: Isola, Freeman, Torralba

Image synthesis

"Duck"

Generator

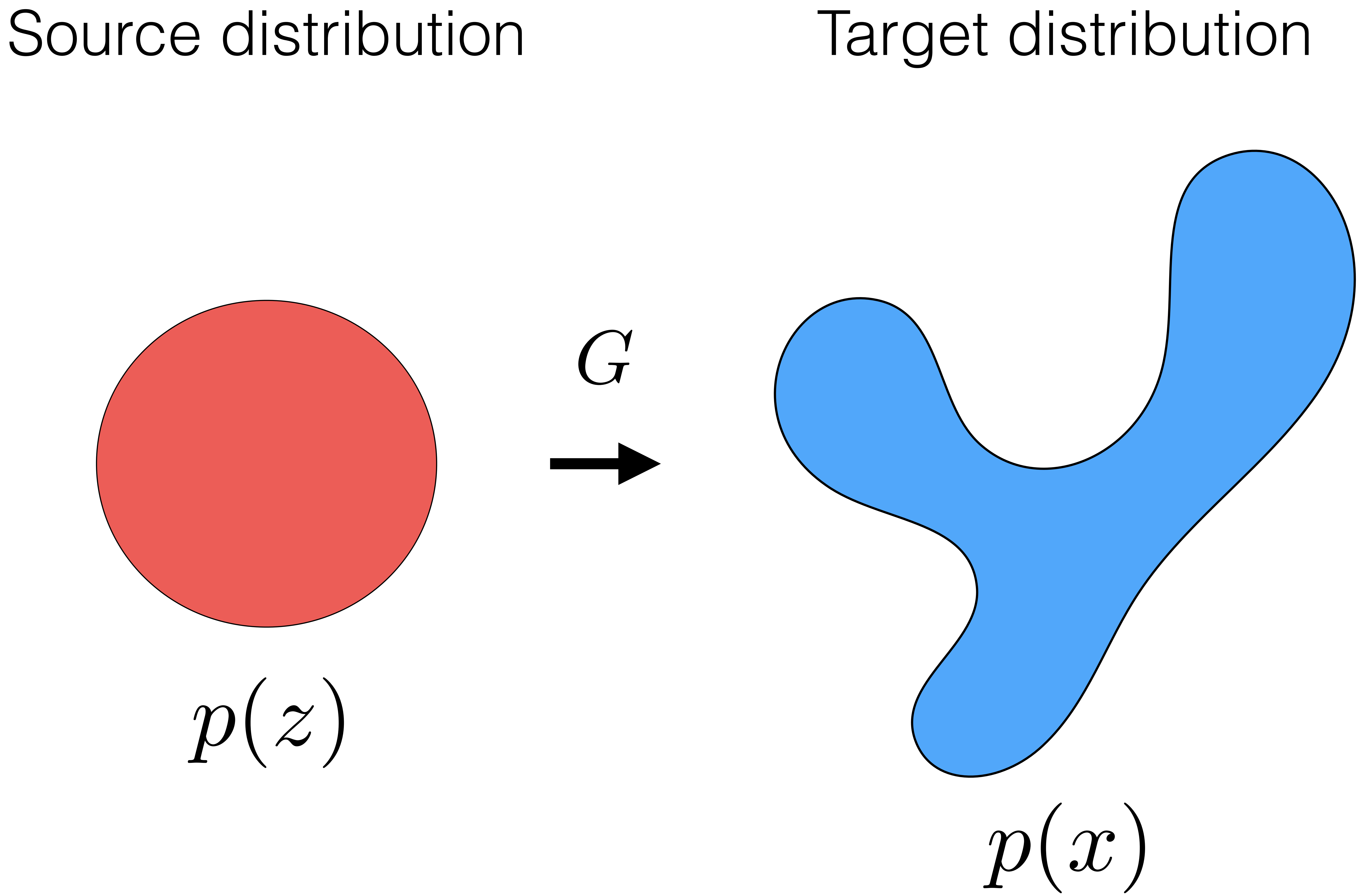


label y

image **x**

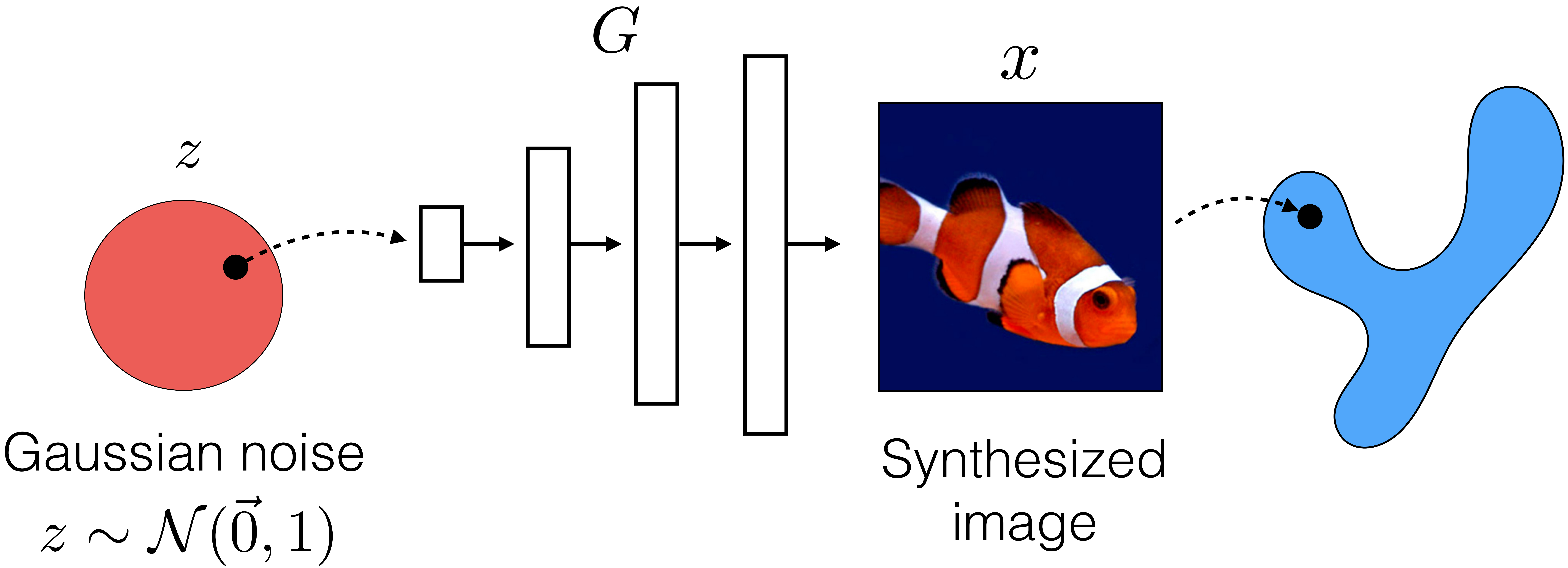
Source: Isola, Freeman, Torralba

Neural networks as distribution transformers



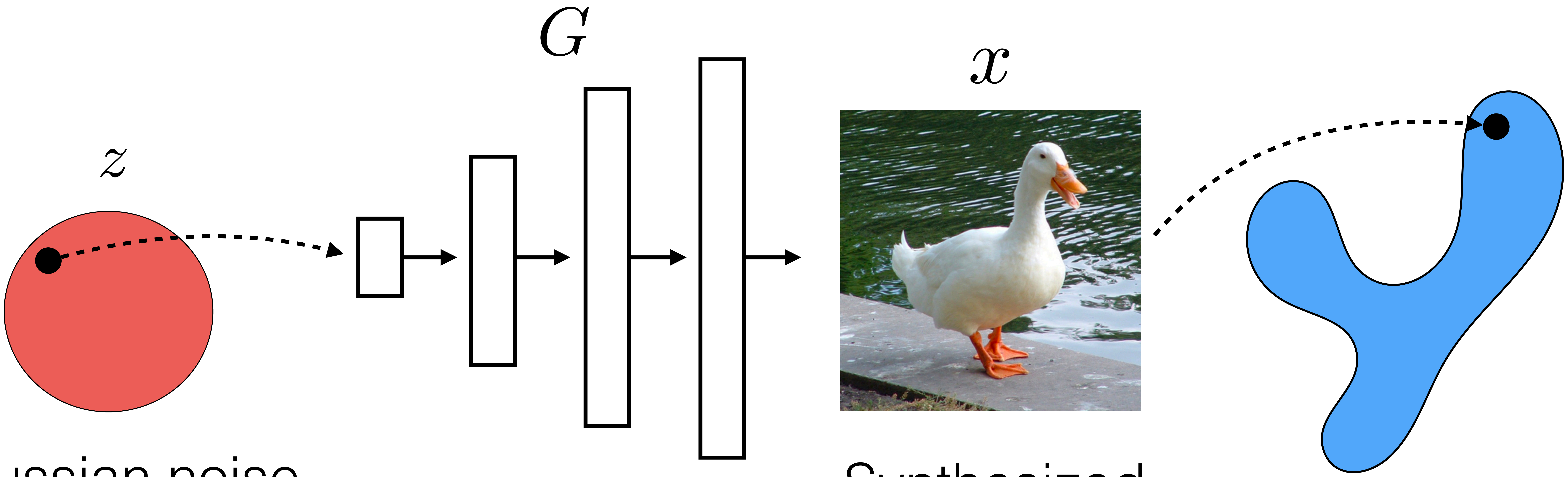
Source: Isola, Freeman, Torralba

Neural networks as distribution transformers



Source: Isola, Freeman, Torralba

Neural networks as distribution transformers



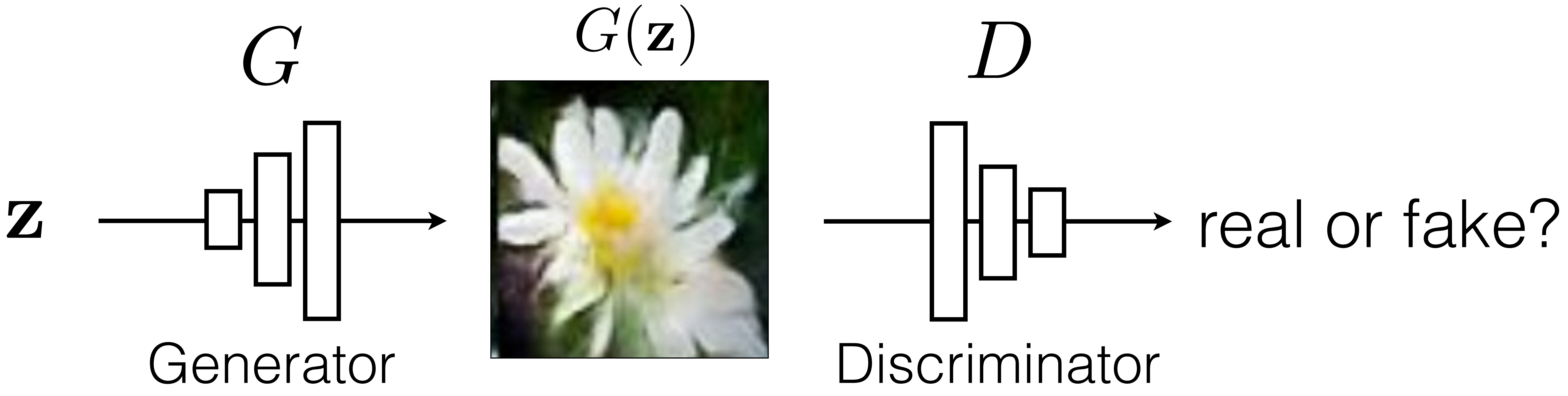
Gaussian noise

$z \sim \mathcal{N}(\vec{0}, 1)$

Synthesized image

Source: Isola, Freeman, Torralba

Generative adversarial networks (GANs)

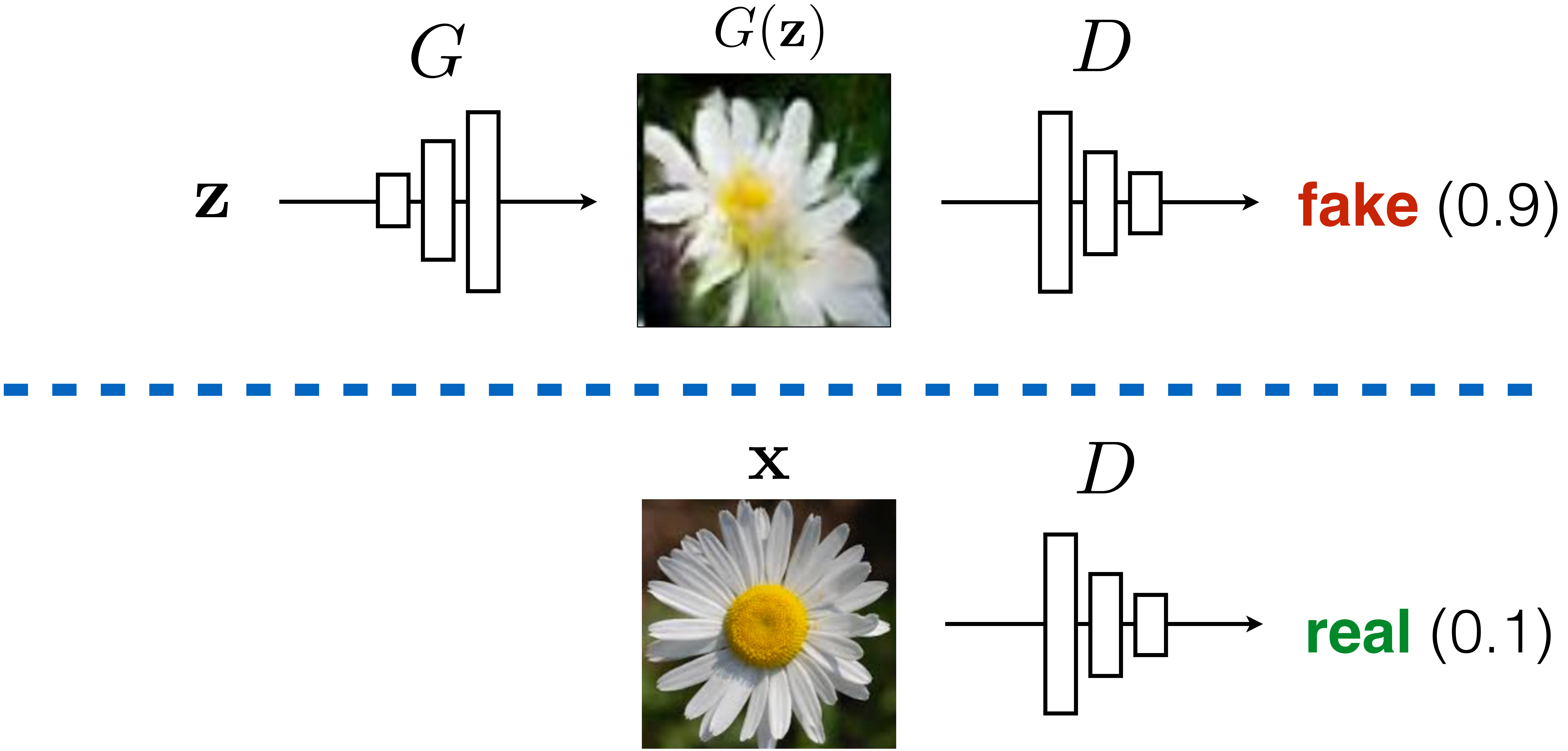


G tries to synthesize fake images that fool **D**

D tries to identify the fakes

[Goodfellow et al., 2014]

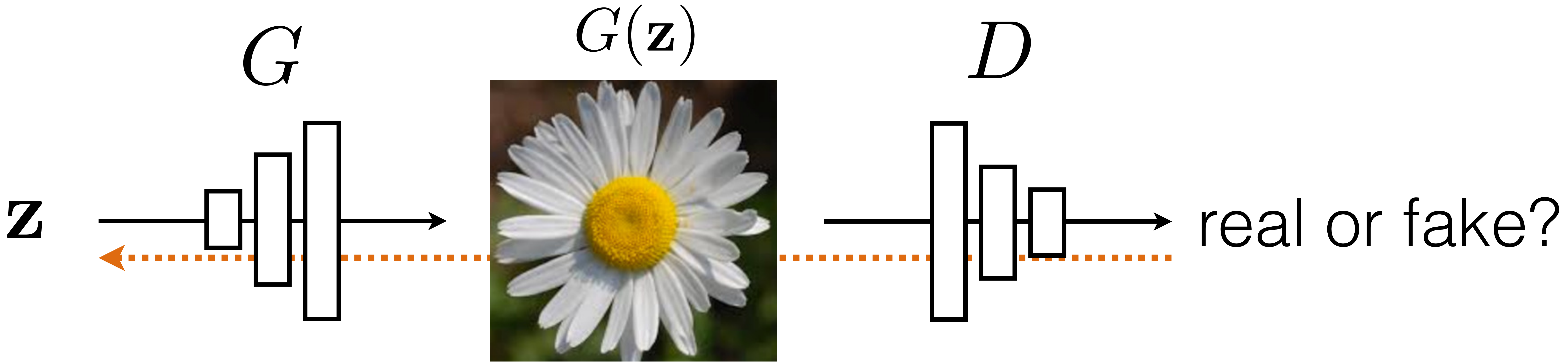
Source: Isola, Freeman, Torralba



$$\arg\max_{D} \mathbb{E}_{\mathbf{z},\mathbf{x}}\left[\ \boxed{\log D(G(\mathbf{z}))} \quad + \quad \boxed{\log\left(1 - D(\mathbf{x})\right)}\ \right]$$

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba

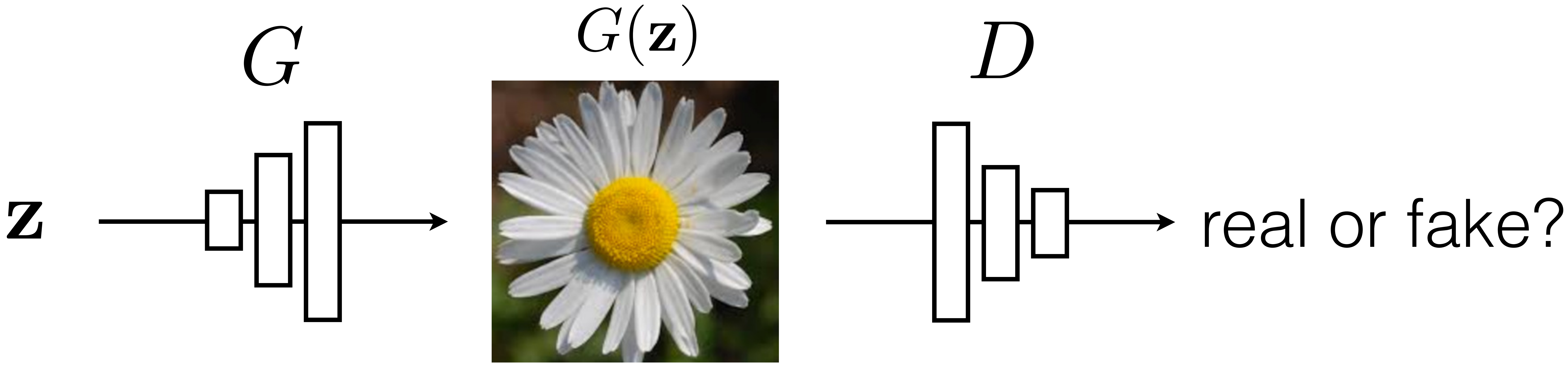


G tries to synthesize fake images that ***fool*** **D**:

$$\arg \boxed{\min_{G}} \; \mathbb{E}_{\mathbf{z},\mathbf{x}}[\; \log D(G(\mathbf{z})) \;+\; \log(1 - D(\mathbf{x})) \;]$$

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba



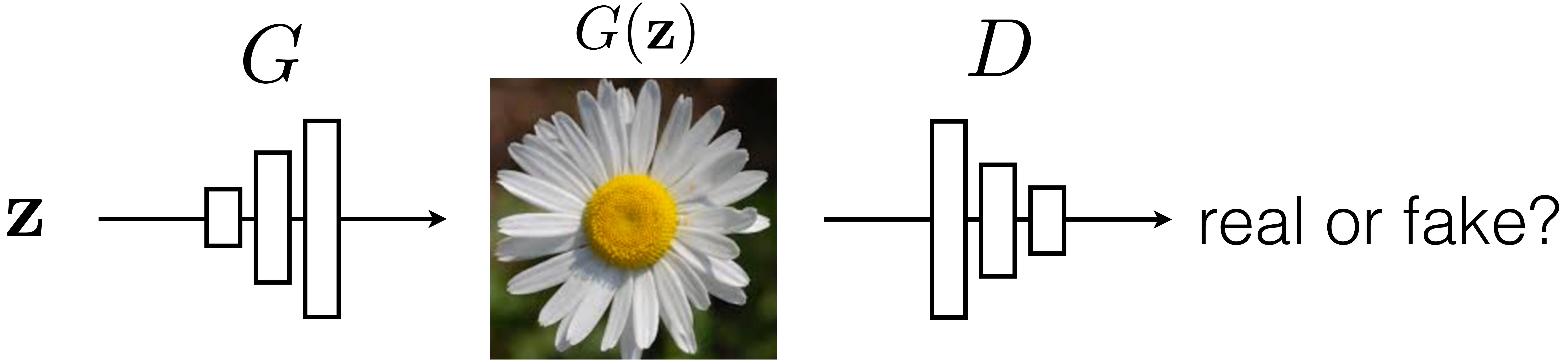
G tries to synthesize fake images that ***fool*** the ***best*** **D**:

$$\arg \boxed{\min_{G}} \boxed{\max_{D}} \; \mathbb{E}_{\mathbf{z},\mathbf{x}}[\; \log D(G(\mathbf{z})) \; + \; \log(1 - D(\mathbf{x})) \;]$$

[Goodfellow et al., 2014]

Source: Isola, Freeman, Torralba

Training



G tries to synthesize fake images that fool **D**

D tries to identify the fakes

- Training: iterate between training D and G with backprop.
- Global optimum when G reproduces data distribution (see book)

[Goodfellow et al., 2014]

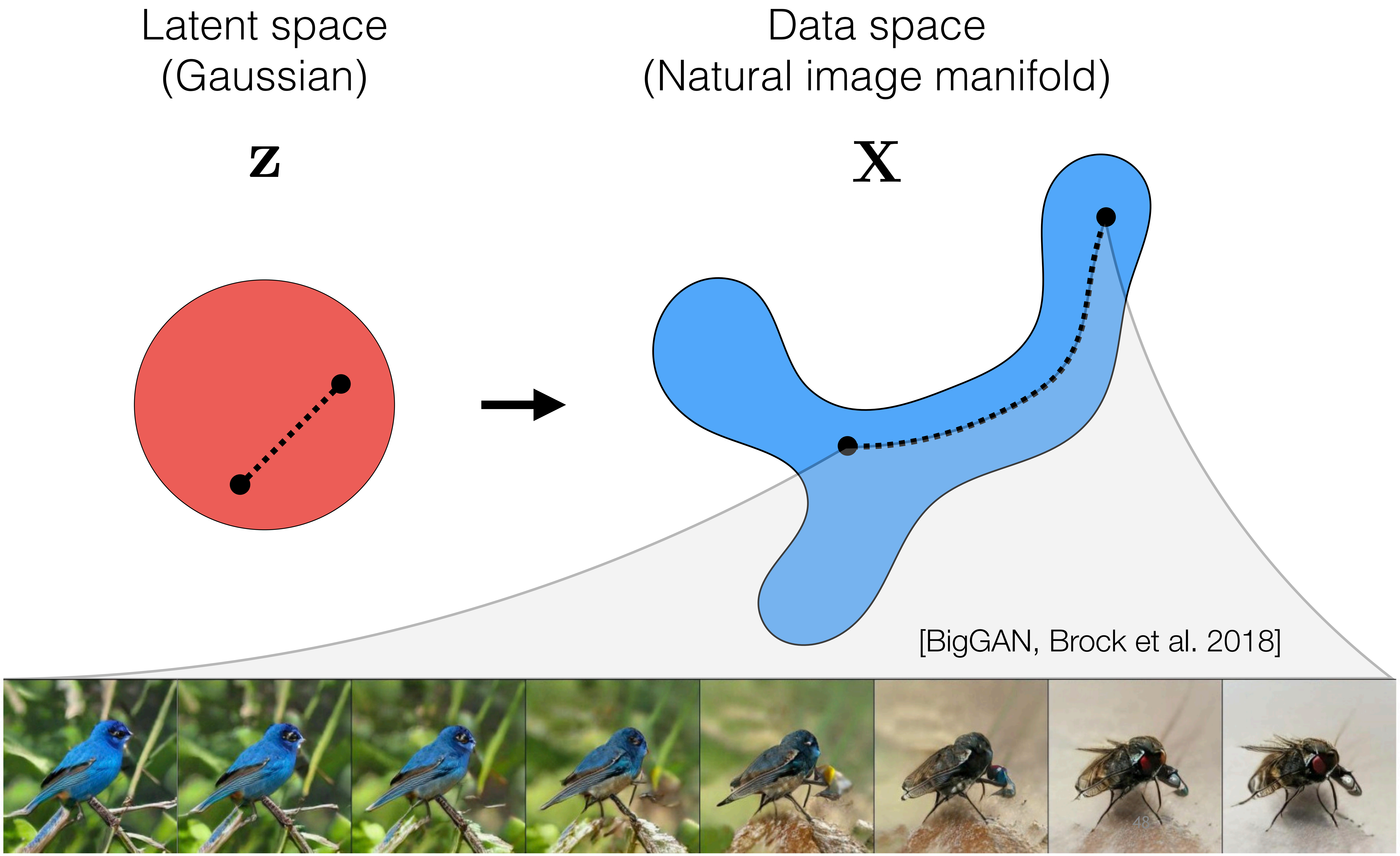
Source: Isola, Freeman, Torralba

Samples from BigGAN

[Brock et al. 2018]

More here: https://arxiv.org/pdf/1809.11096.pdf

Source: Isola, Freeman, Torralba


Latent space
(Gaussian)
$\mathbf{z}$
Data space
(Natural image manifold)
$\mathbf{x}$
[BigGAN, Brock et al. 2018]

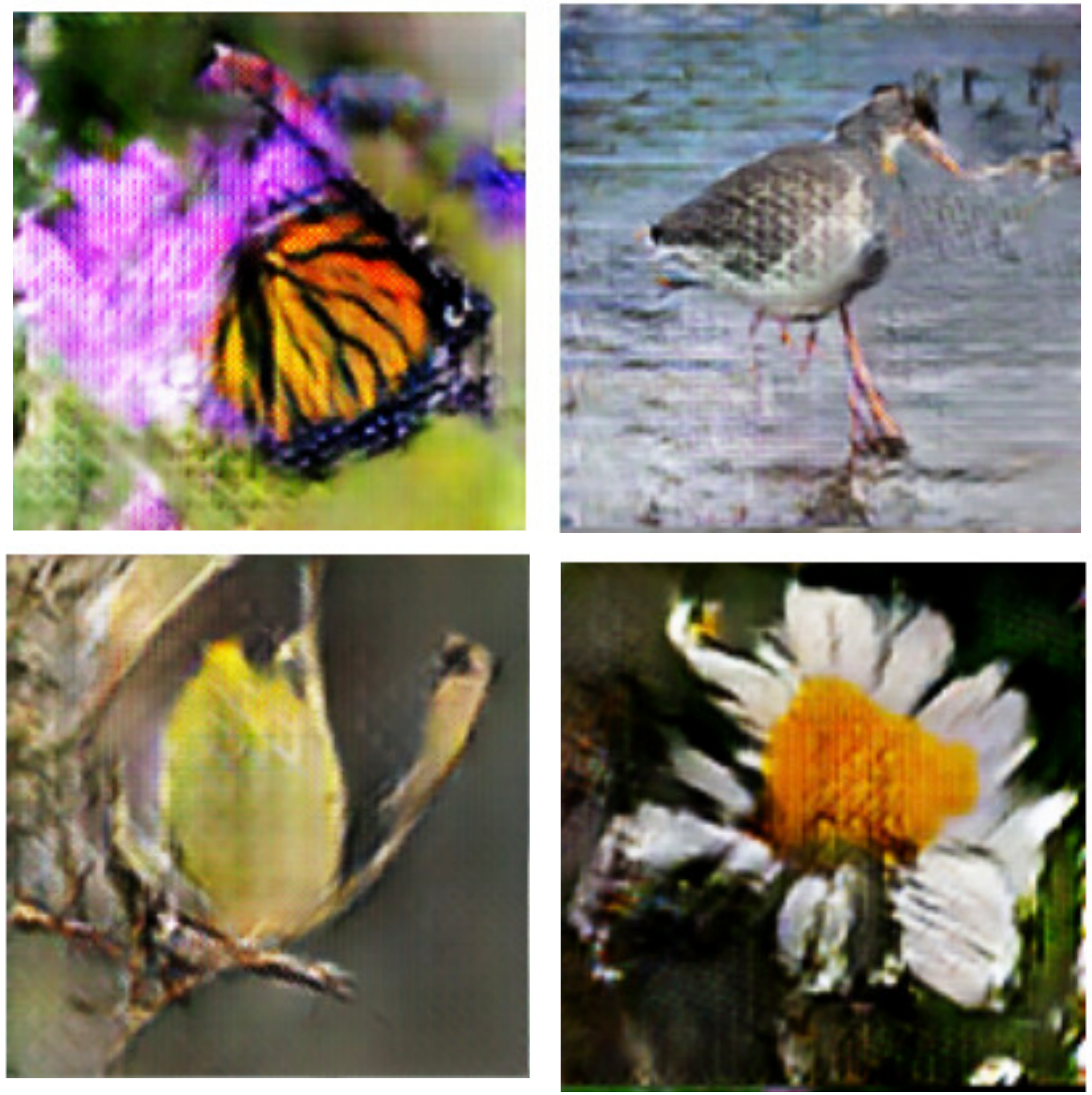
https://github.com/NVlabs/stylegan3 [Karras et al., “Alias-Free Generative Adversarial Networks”, 2021]

[Karras et al., “Alias-Free Generative Adversarial Networks”, 2021]

[Karras et al., “Alias-Free Generative Adversarial Networks”, 2021]

Rapid progress due mostly to better architectures

ACGAN [Odena et al. 2016]



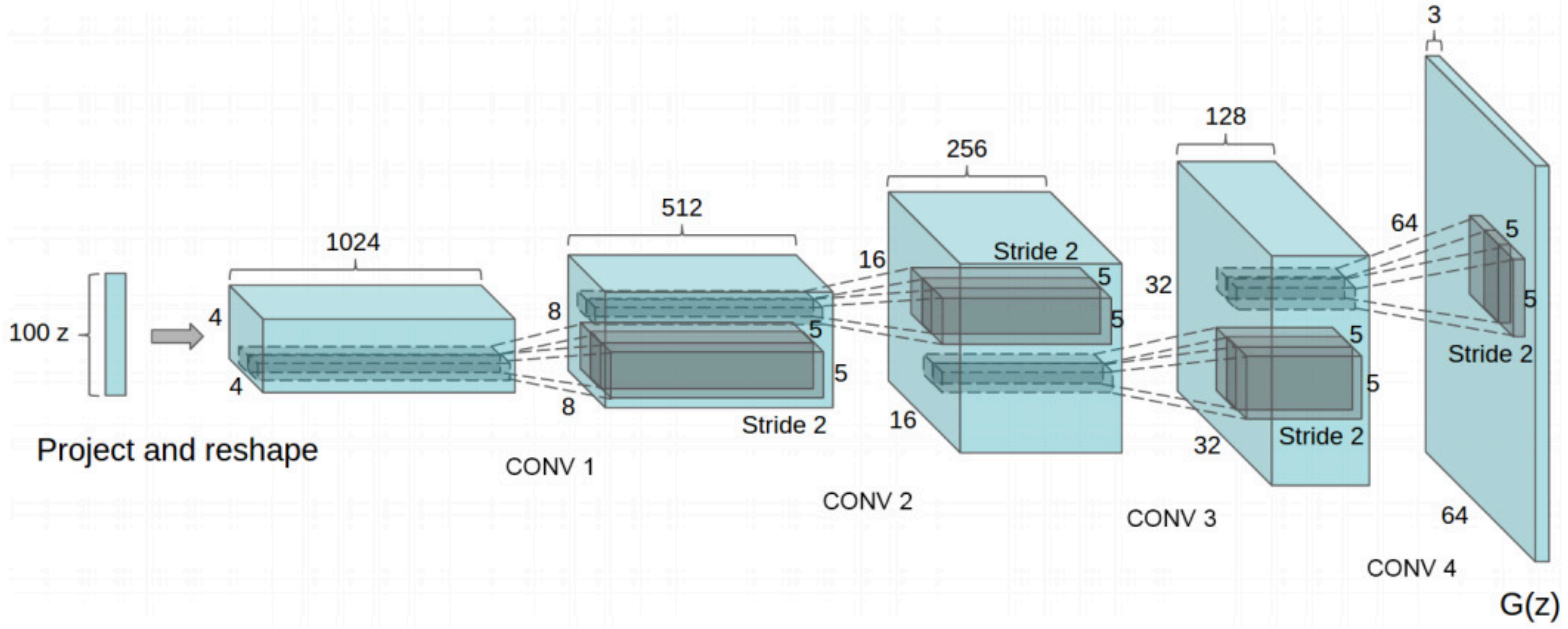
BigGAN [Brock et al. 2018]

Both trained on ImageNet

Source: Isola, Freeman, Torralba

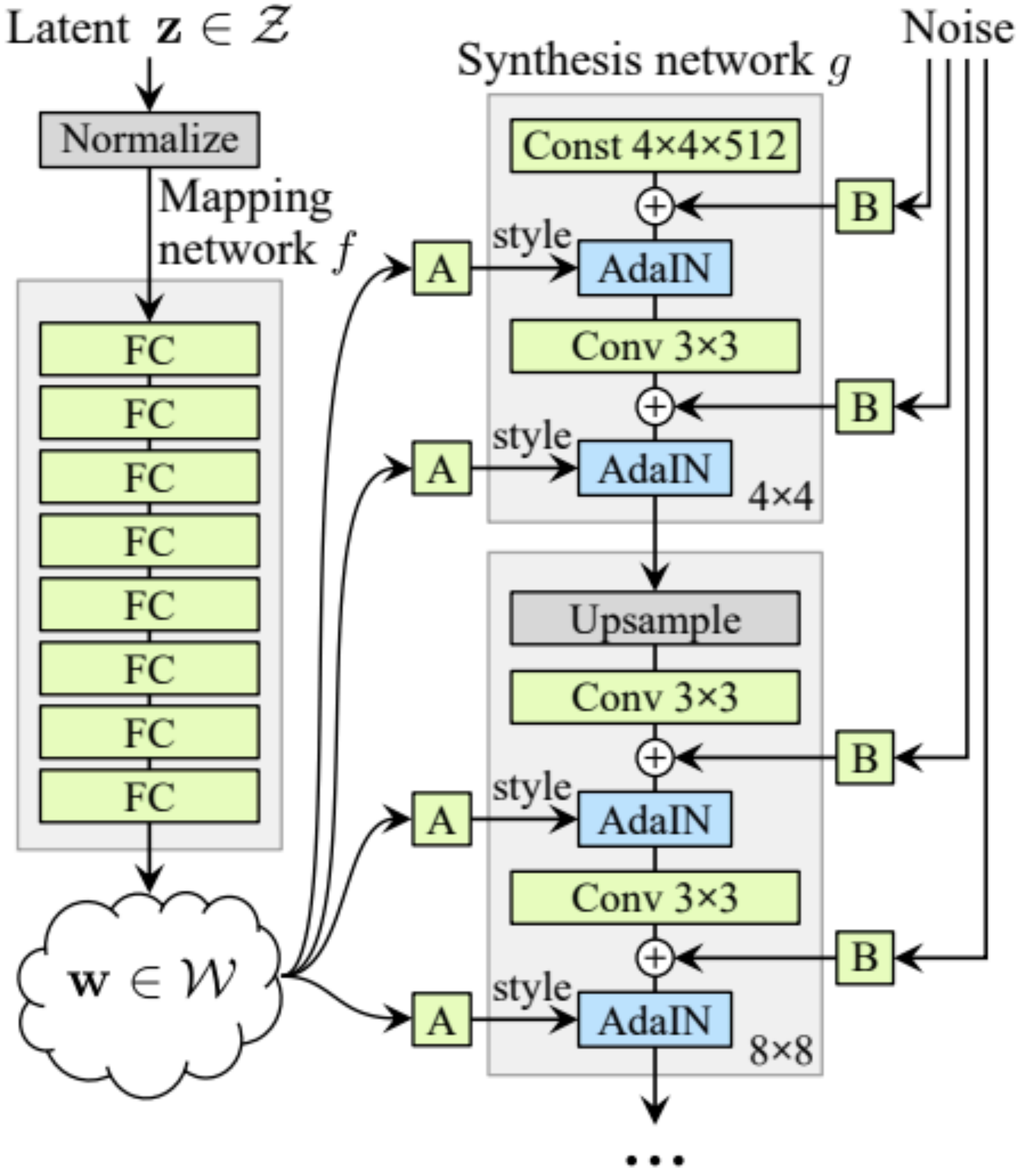
Architectures

DCGAN
[Radford, Metz, Chintala 2016]



Transpose convolution + batch norm + nonlinearities

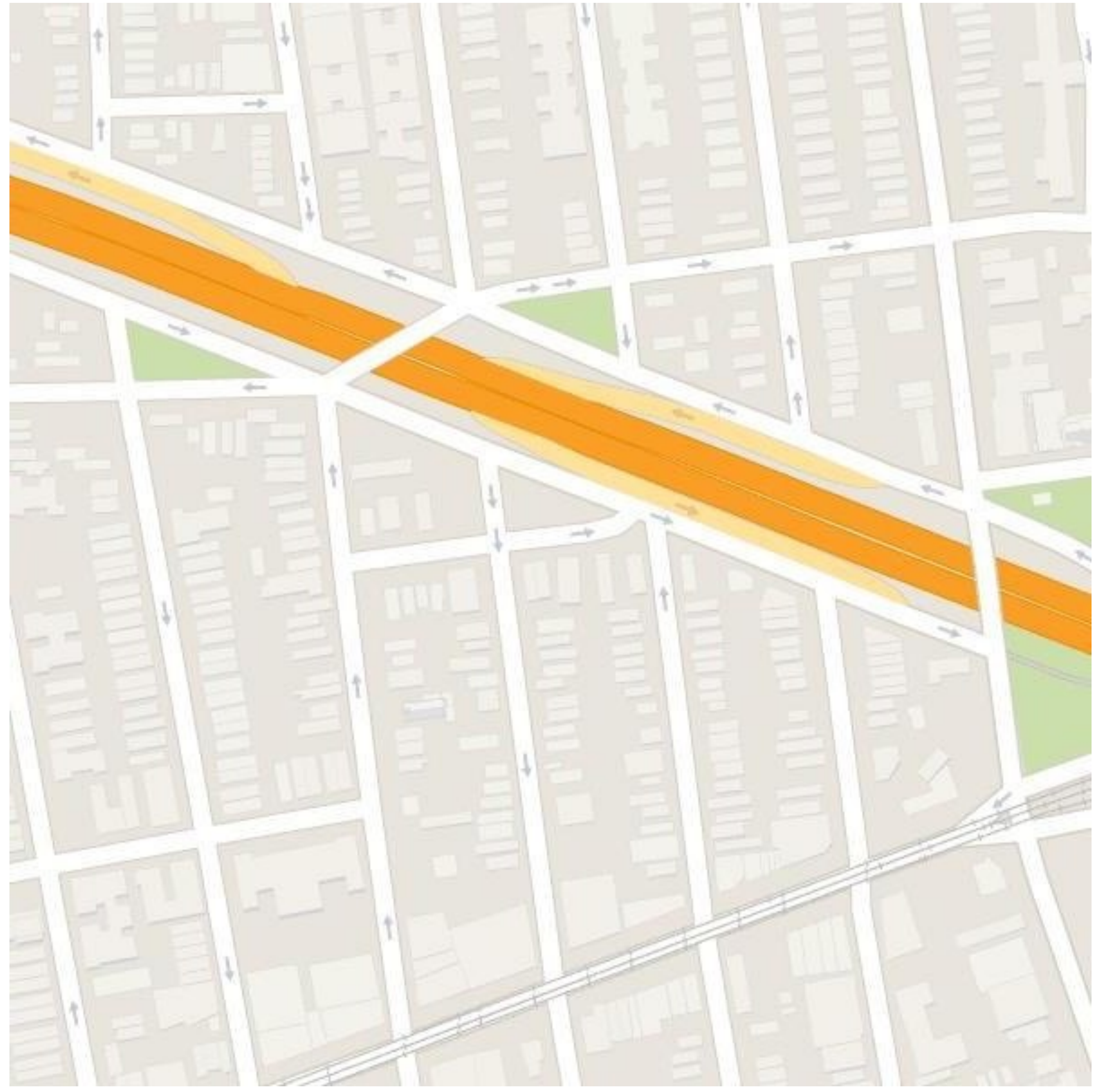
StyleGAN
[Karras, Laine, Aila 2019]



Similar but bigger and lots of engineering details.

Source: Isola, Freeman, Torralba

Image translation



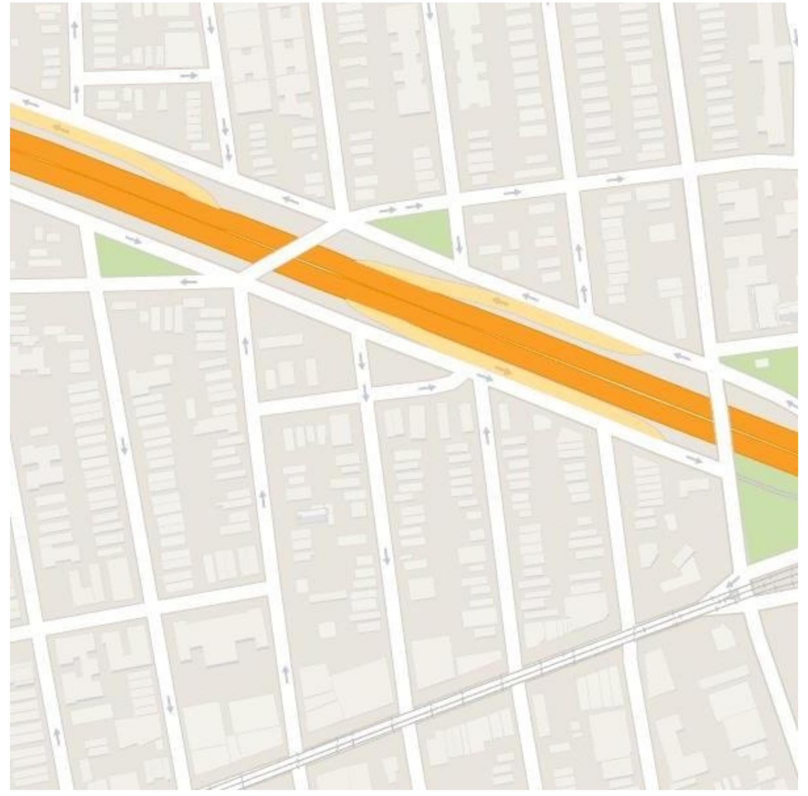
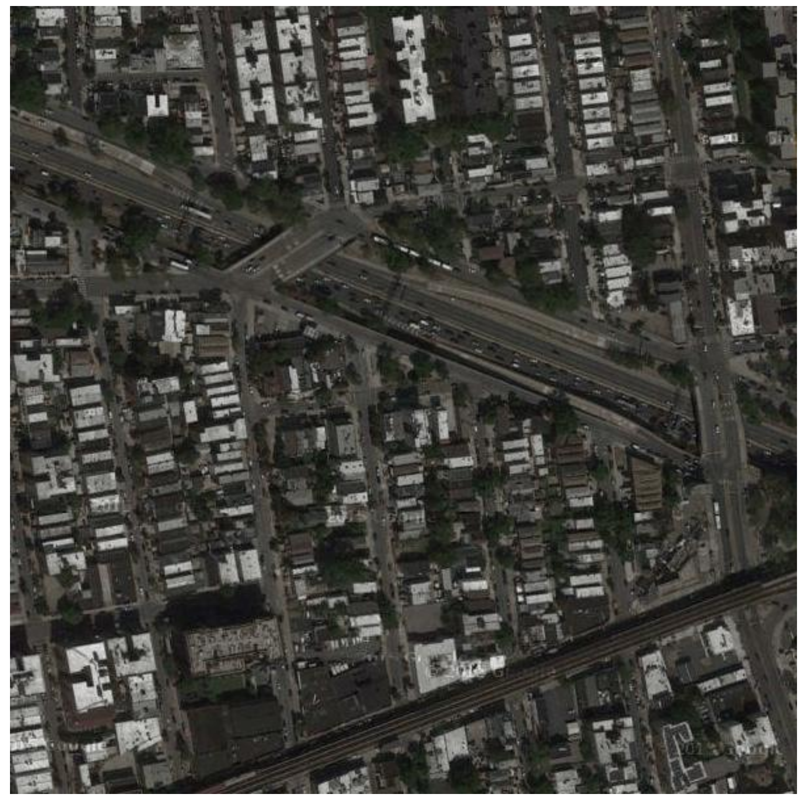
Google Map

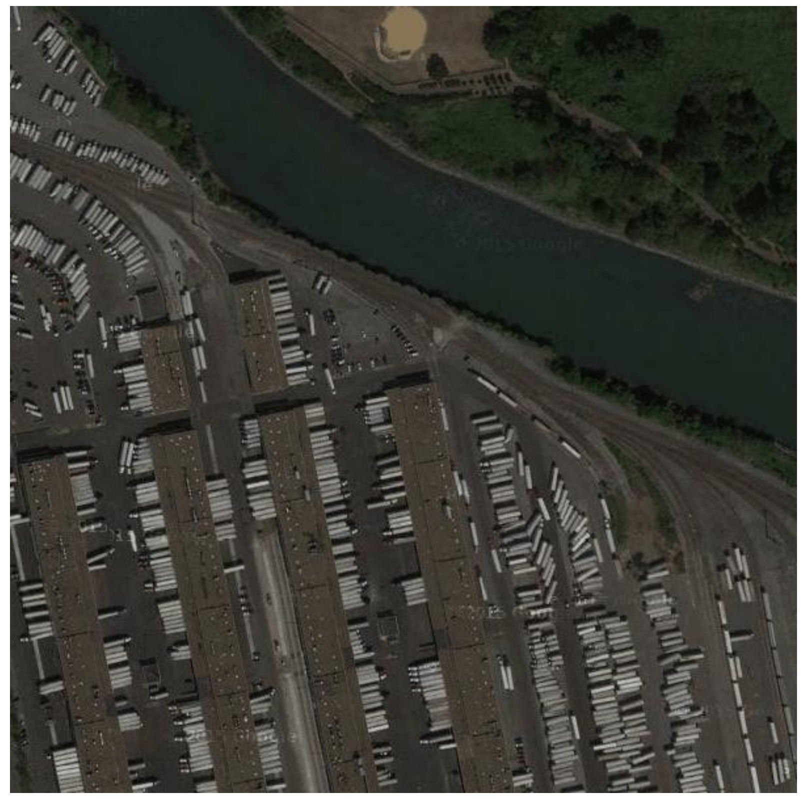
Satellite photo

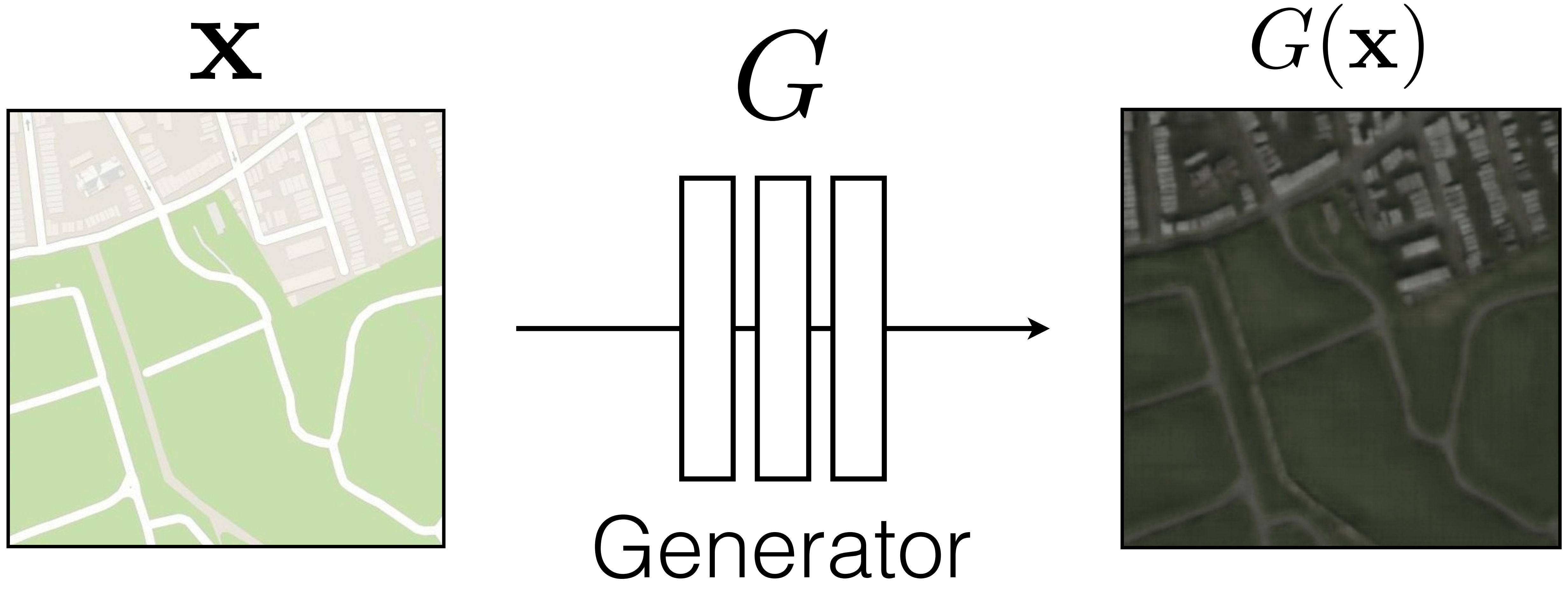
Source: Isola, Freeman, Torralba

Map2Sat

$\mathbf{x}$ $\mathbf{y}$

{  ,  }

{  ,  }

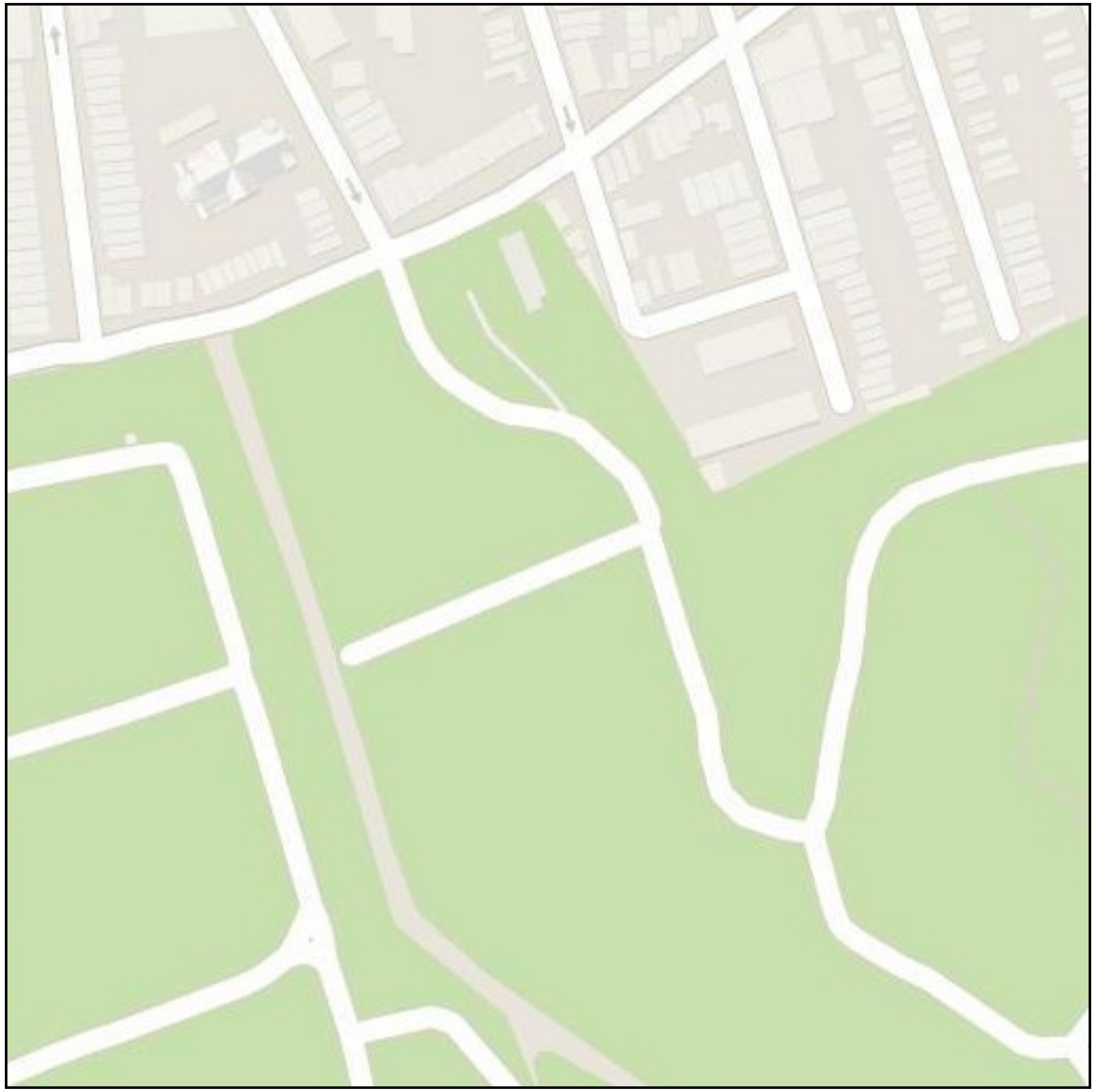


Idea: L1 loss

$$||G(\mathbf{x}) - \mathbf{y}||_1$$

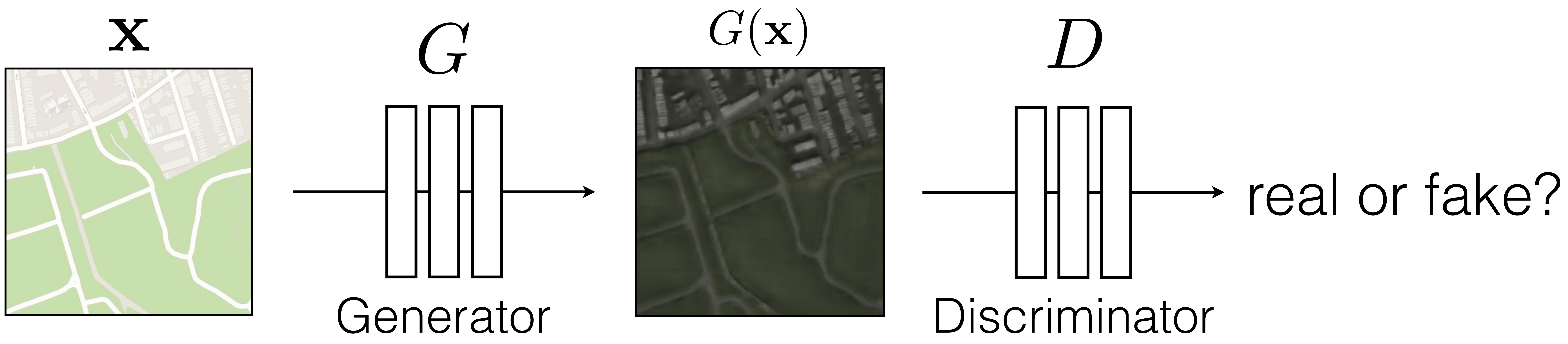
Source: Isola, Freeman, Torralba

Input



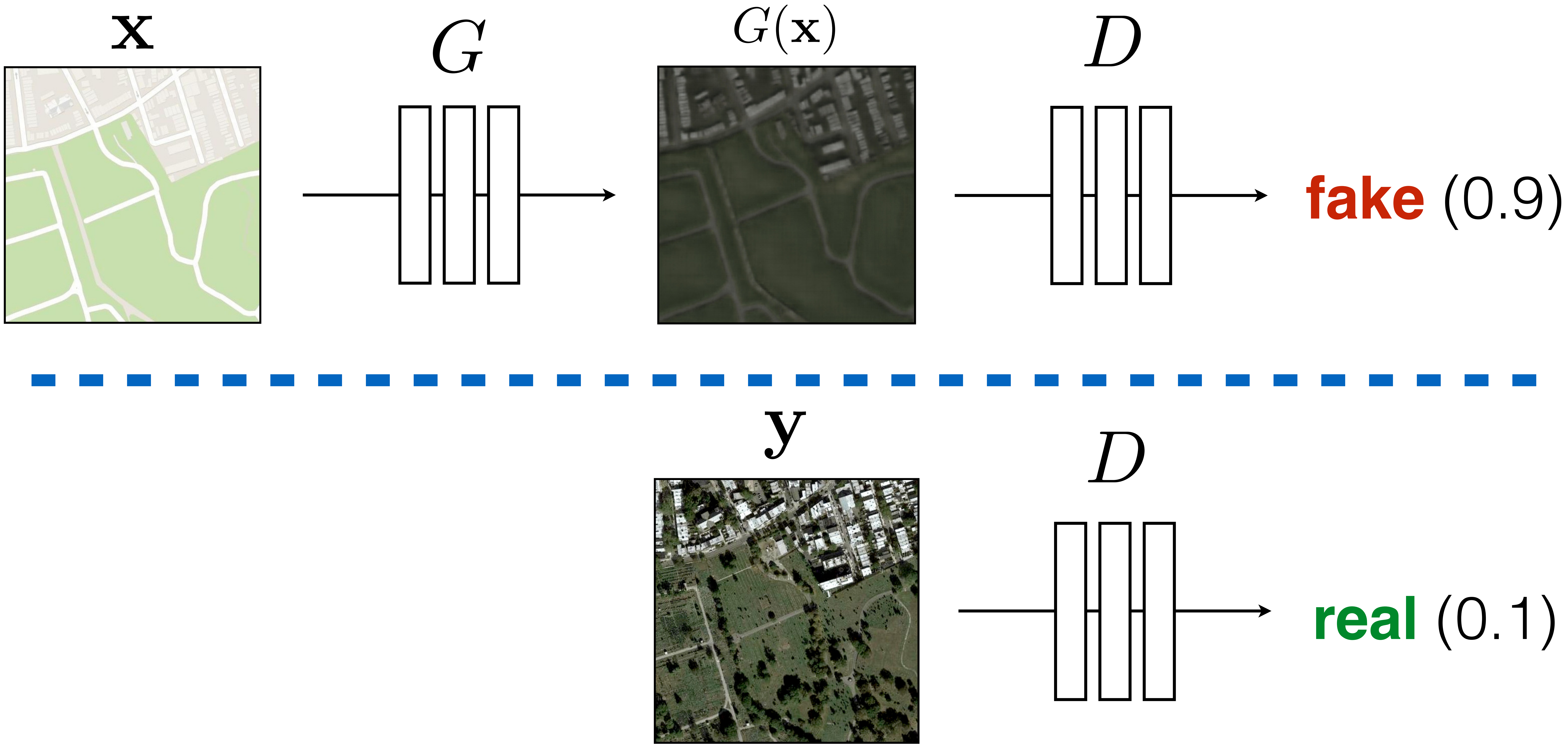
L1 loss

Source: Isola, Freeman, Torralba



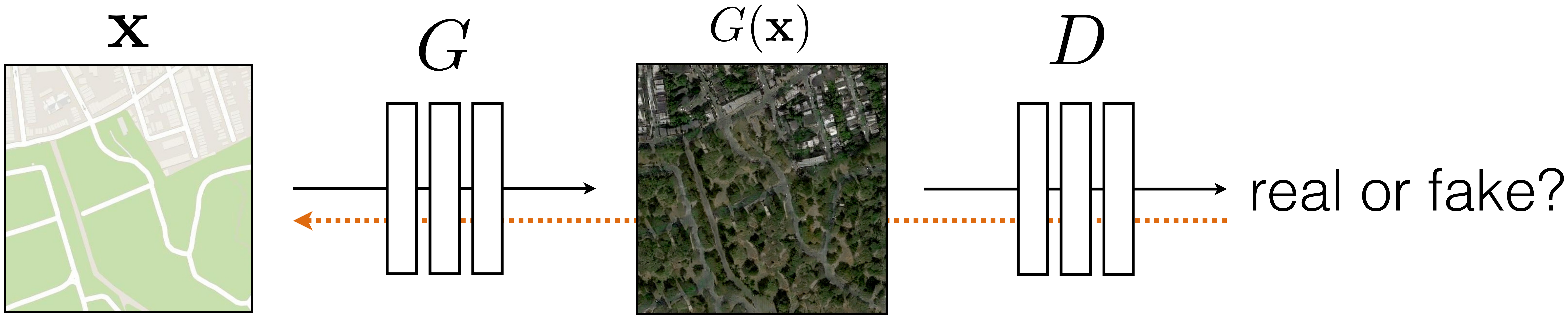
G tries to synthesize fake images that fool **D**

D tries to identify the fakes



$$\arg\max_{D} \; \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\boxed{\log D(G(\mathbf{x}))} + \boxed{\log(1 - D(\mathbf{y}))} \right]$$

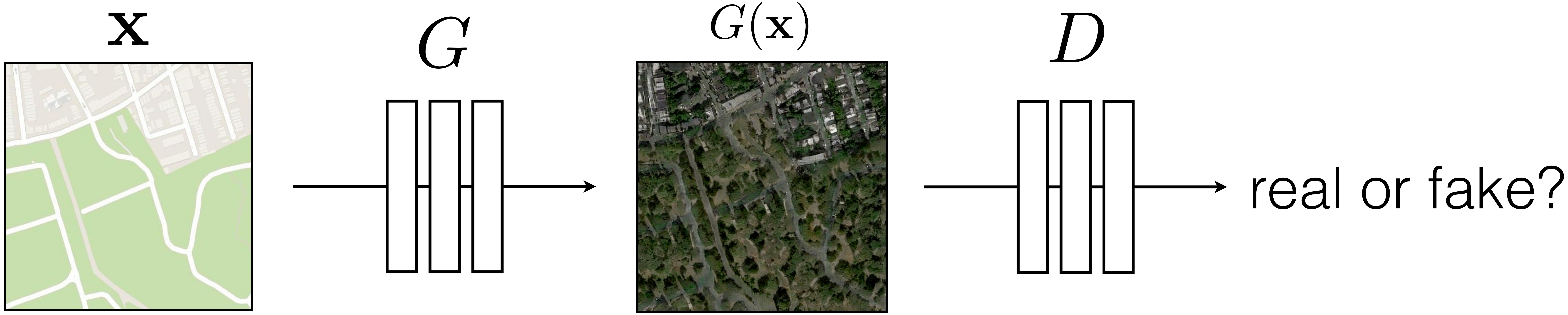
Source: Isola, Freeman, Torralba



G tries to synthesize fake images that ***fool*** **D**:

$$\arg\boxed{\min_{G}}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ \ ]$$

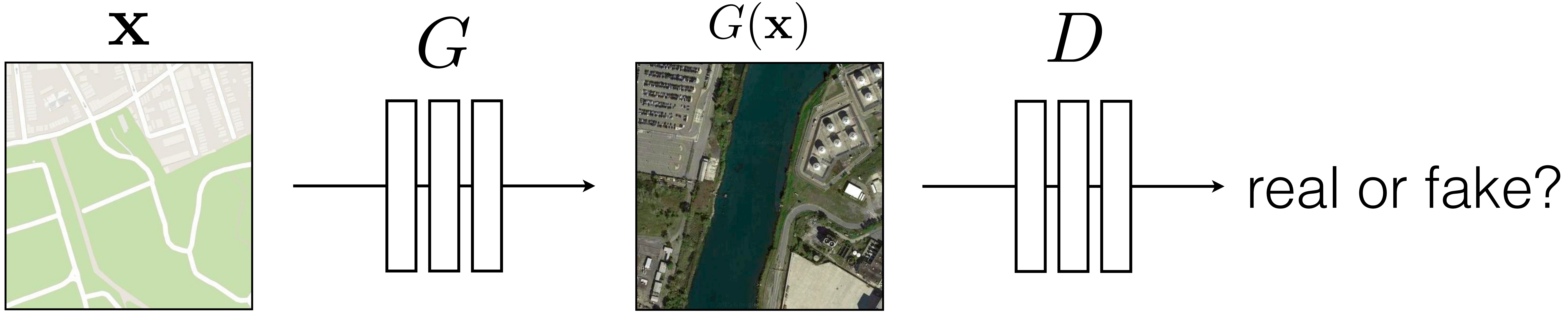
Source: Isola, Freeman, Torralba



G tries to synthesize fake images that ***fool*** the ***best*** **D**:

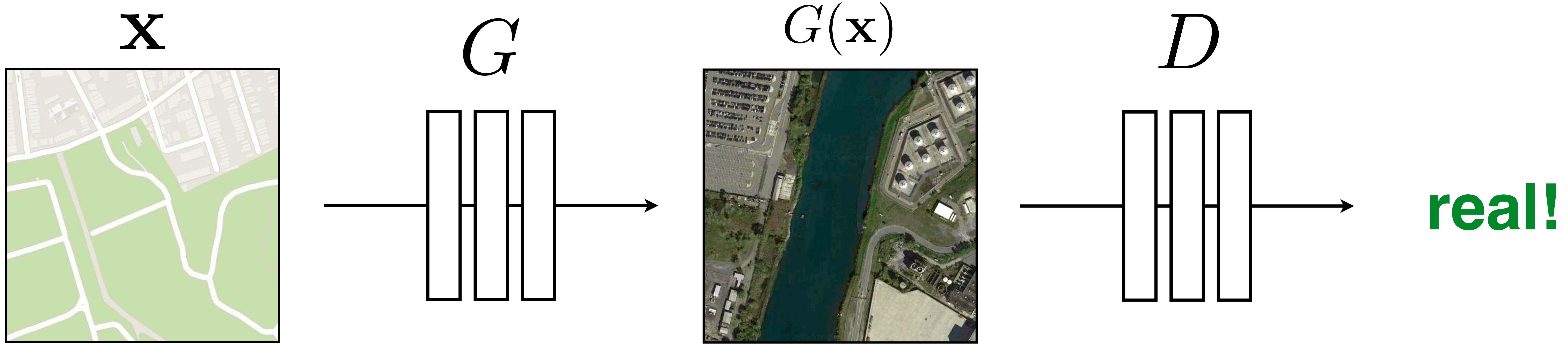
$$\arg \boxed{\min_G} \boxed{\max_D} \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \;+\; \log(1 - D(\mathbf{y})) \;]$$

Source: Isola, Freeman, Torralba



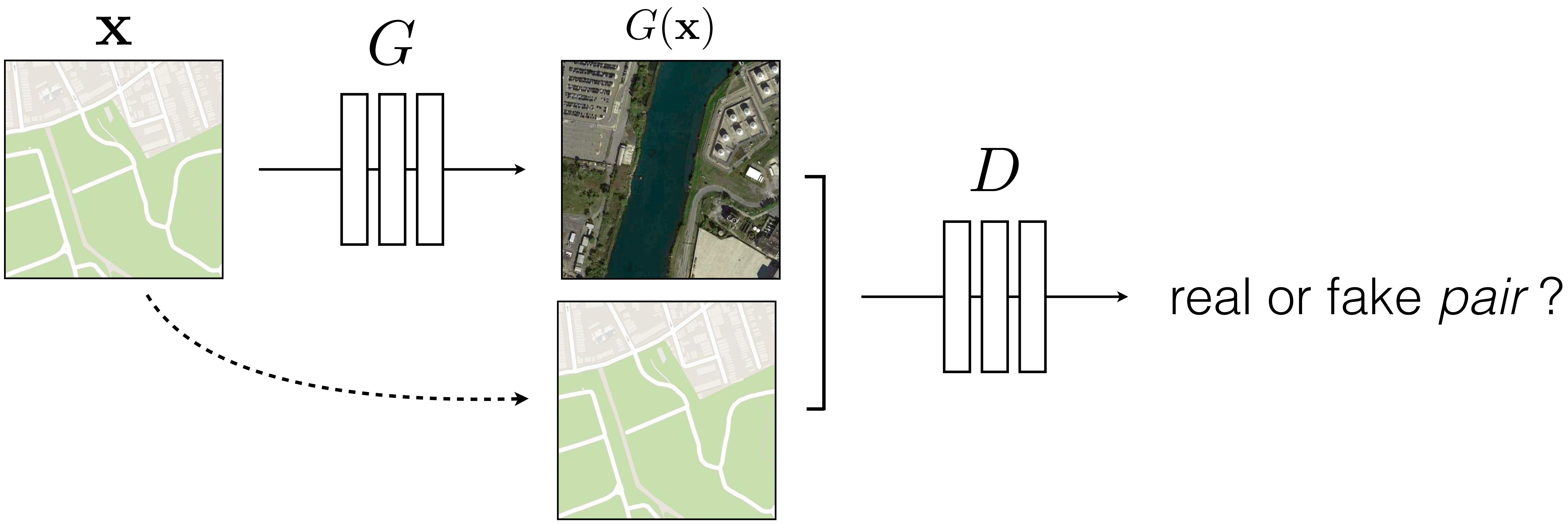
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ \ ]$$

Source: Isola, Freeman, Torralba



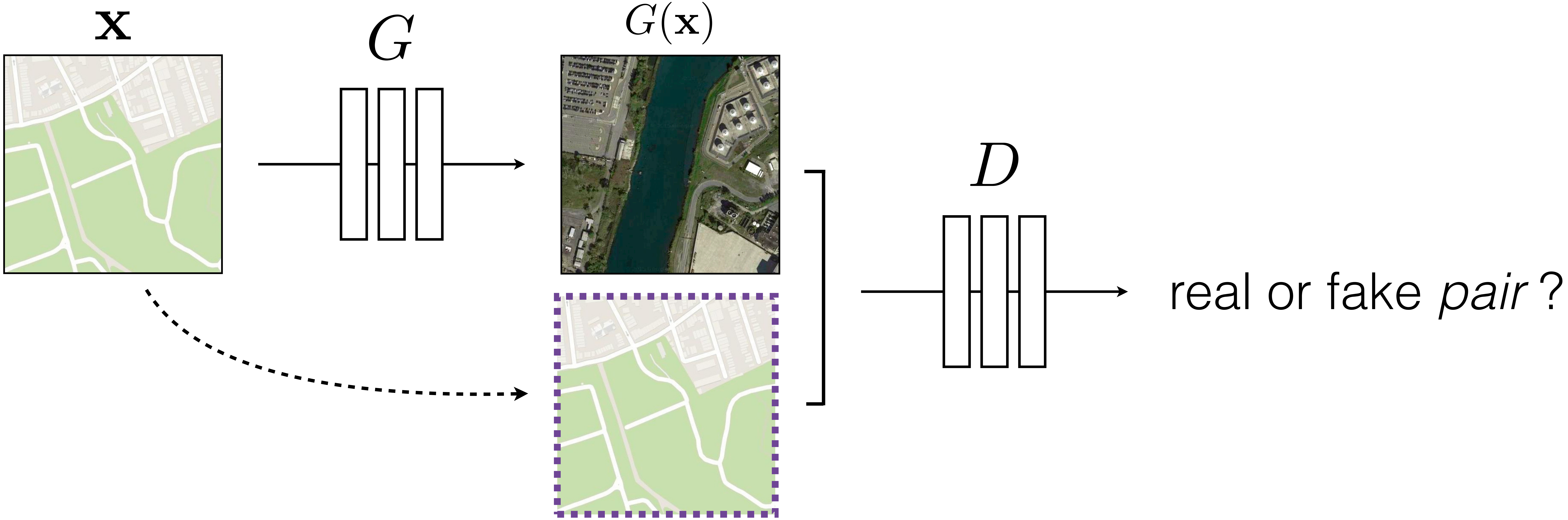
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ \ ]$$

Source: Isola, Freeman, Torralba



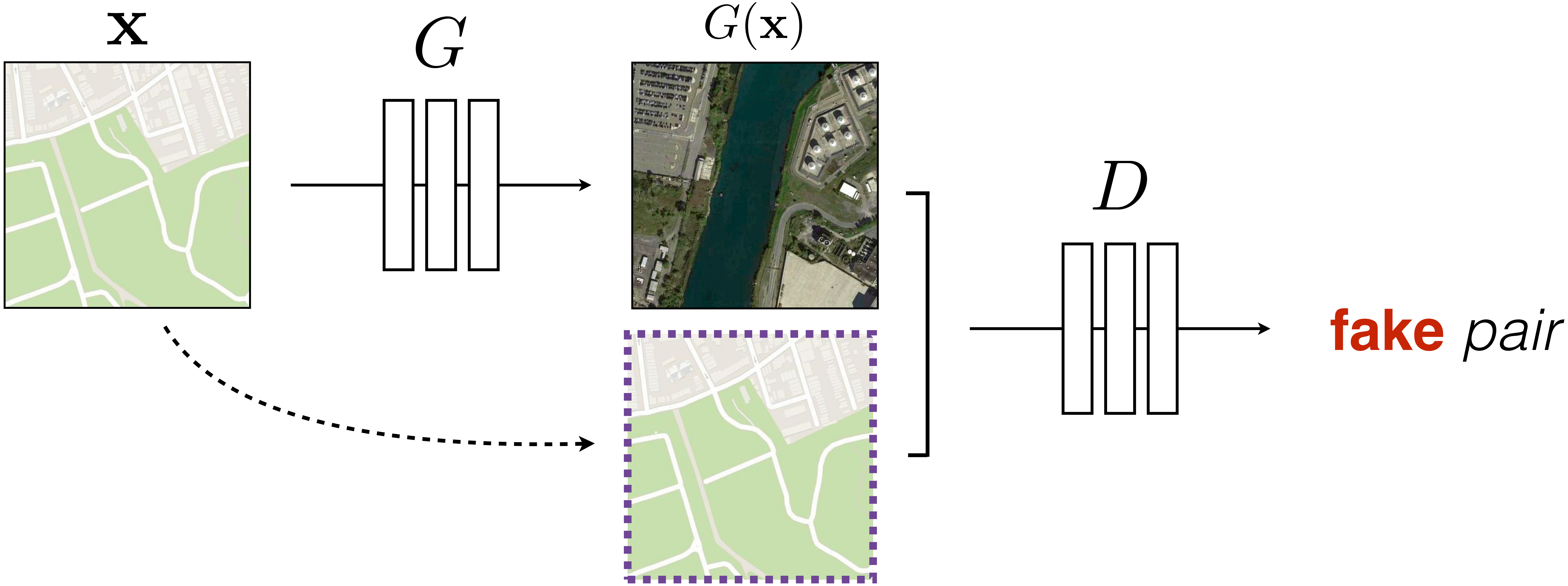
$$\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\ ]$$

Source: Isola, Freeman, Torralba



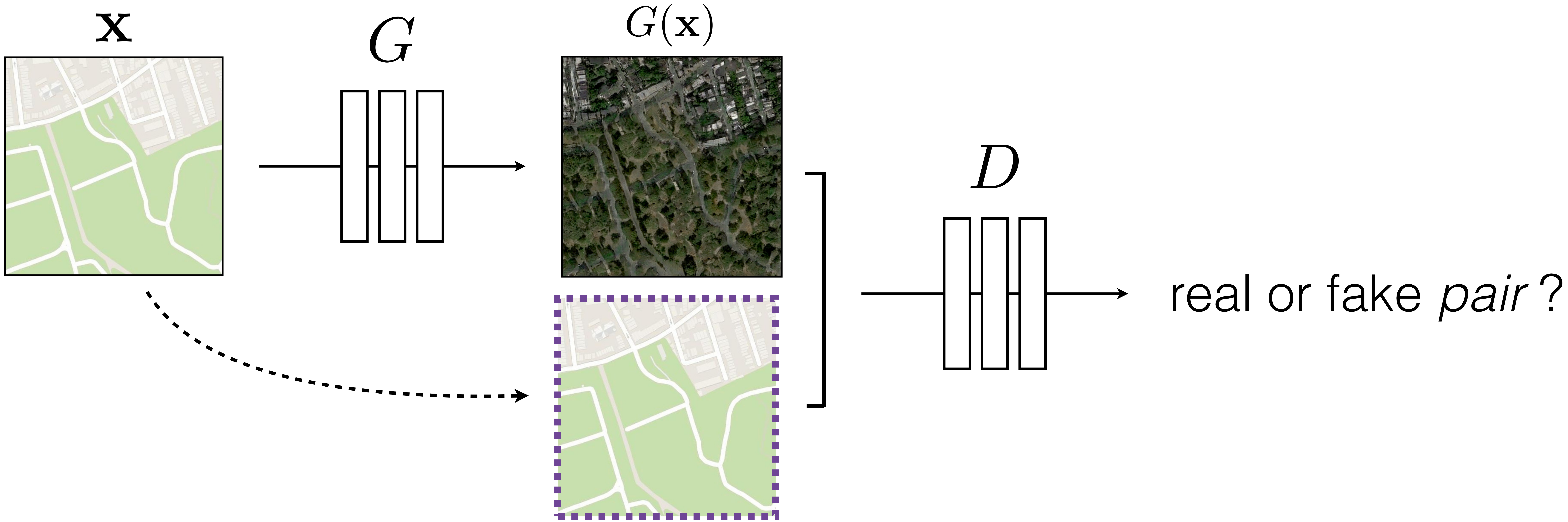
$$\arg\min_{G}\max_{D} \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y})) \;]$$

Source: Isola, Freeman, Torralba



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y})) \;]$$

Source: Isola, Freeman, Torralba



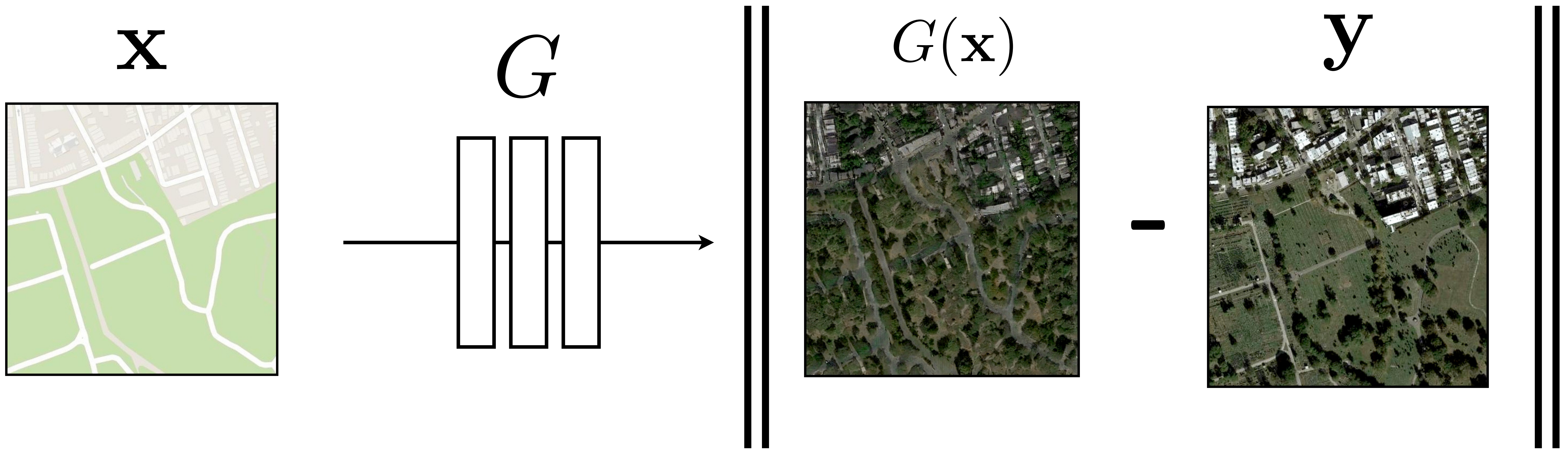
$$\boxed{\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x},\mathbf{y}))\ ]}$$

Source: Isola, Freeman, Torralba

Training Details: Loss function

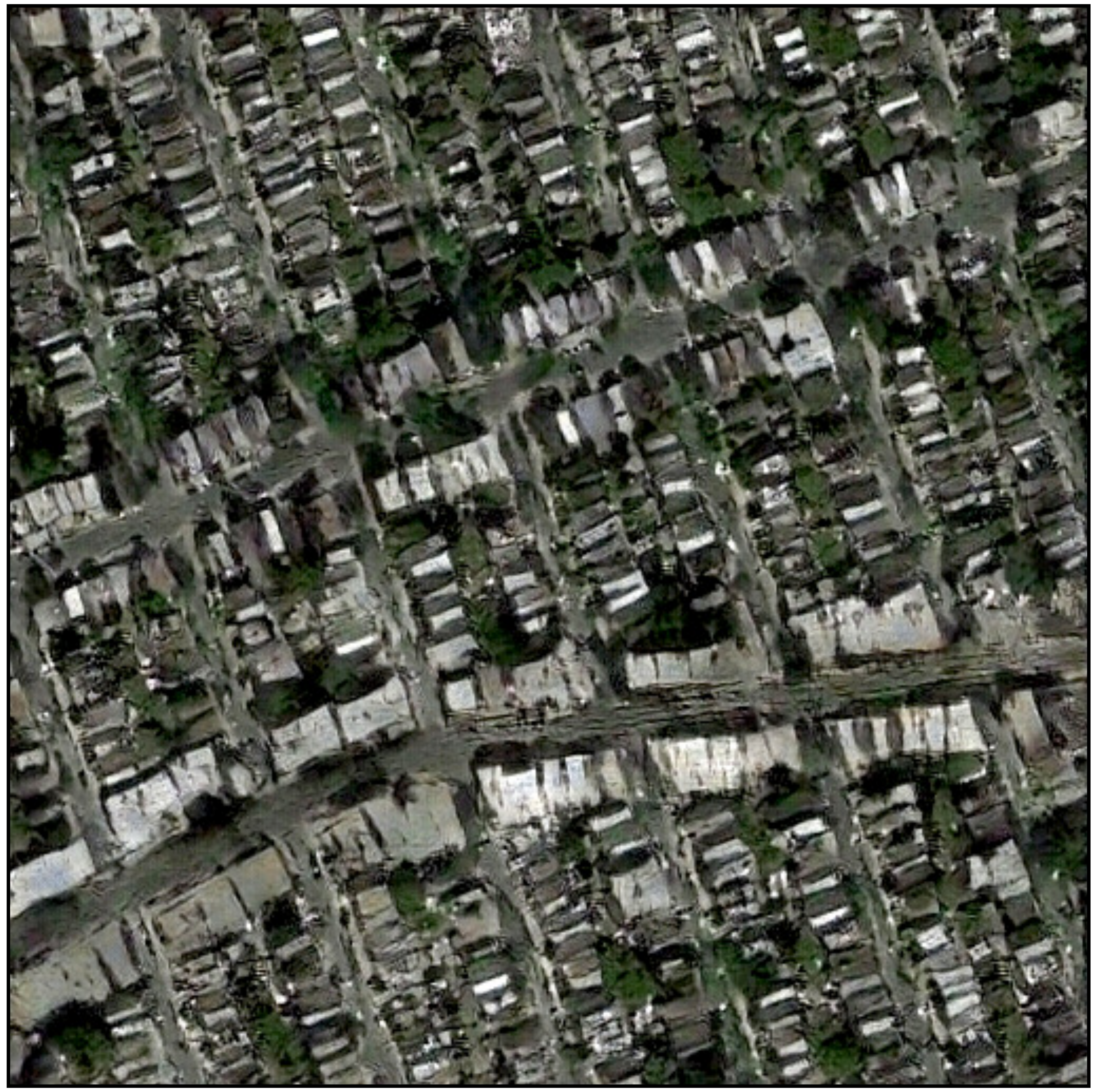
Conditional GAN

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$



Helps stabilize training + faster convergence

Source: Isola, Freeman, Torralba



Data from [maps.google.com]

Source: Isola, Freeman, Torralba

Input

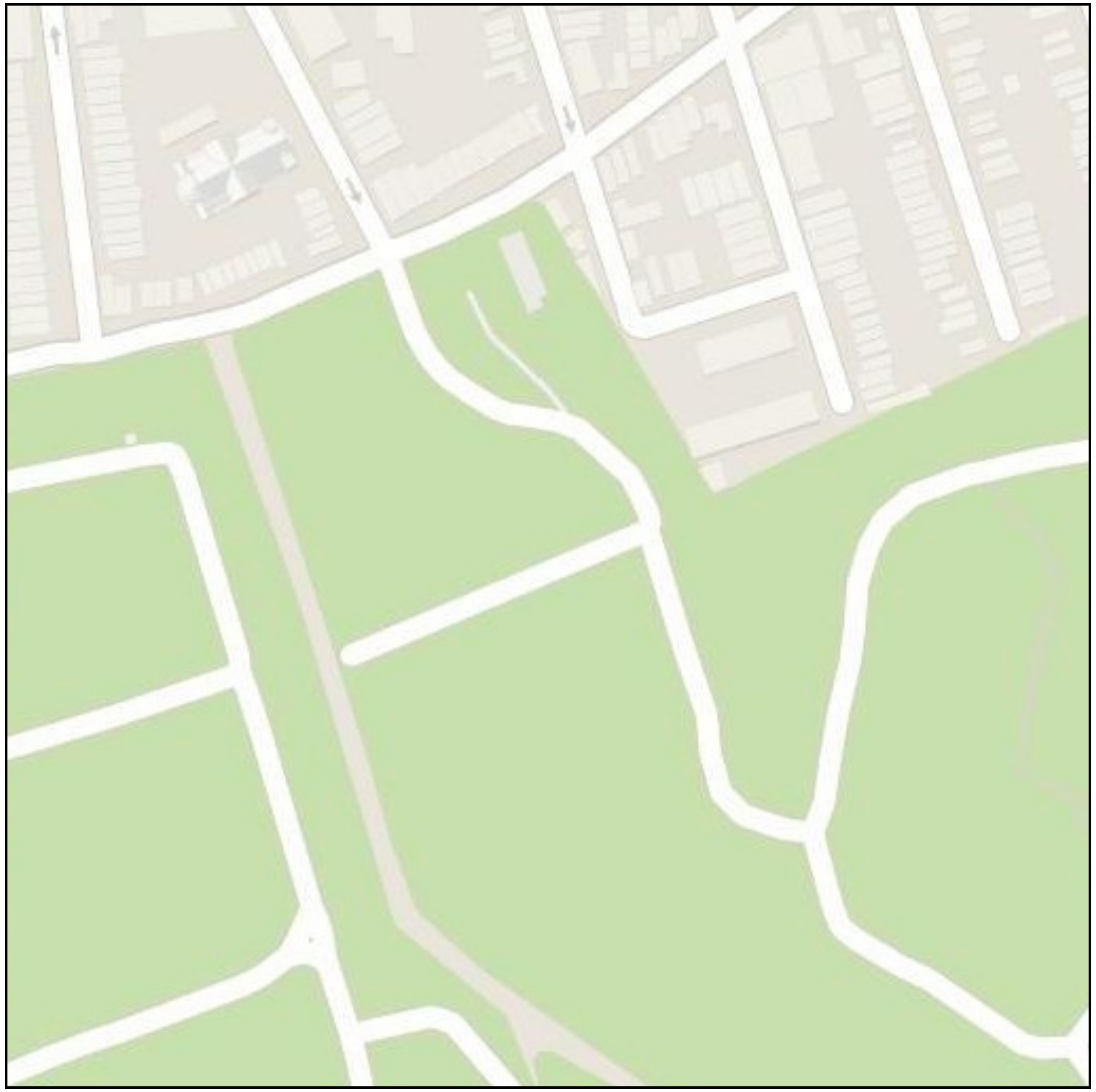
Output

Groundtruth

Data from [maps.google.com]

Source: Isola, Freeman, Torralba

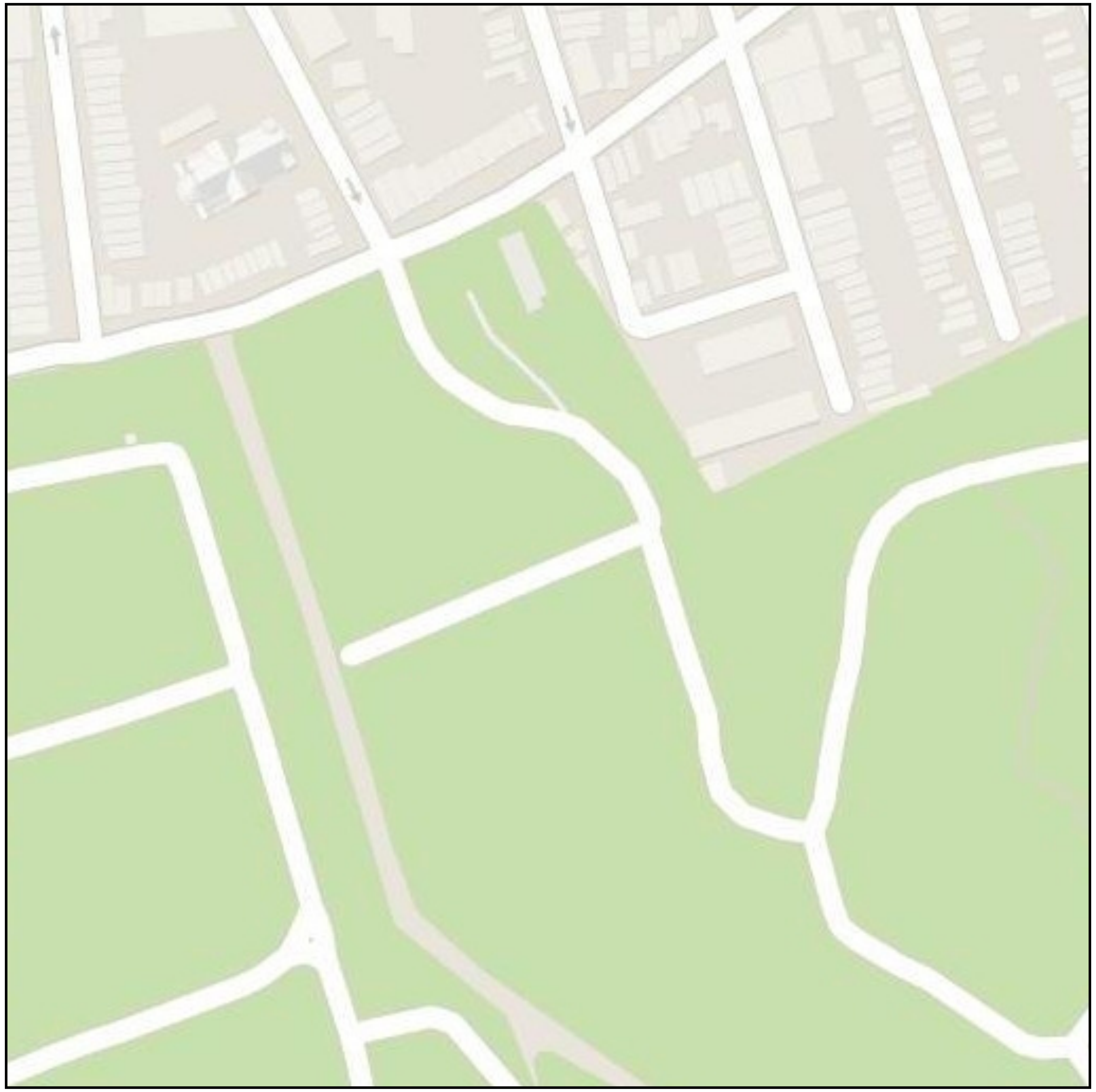
Input



L1 loss only

Source: Isola, Freeman, Torralba

Input

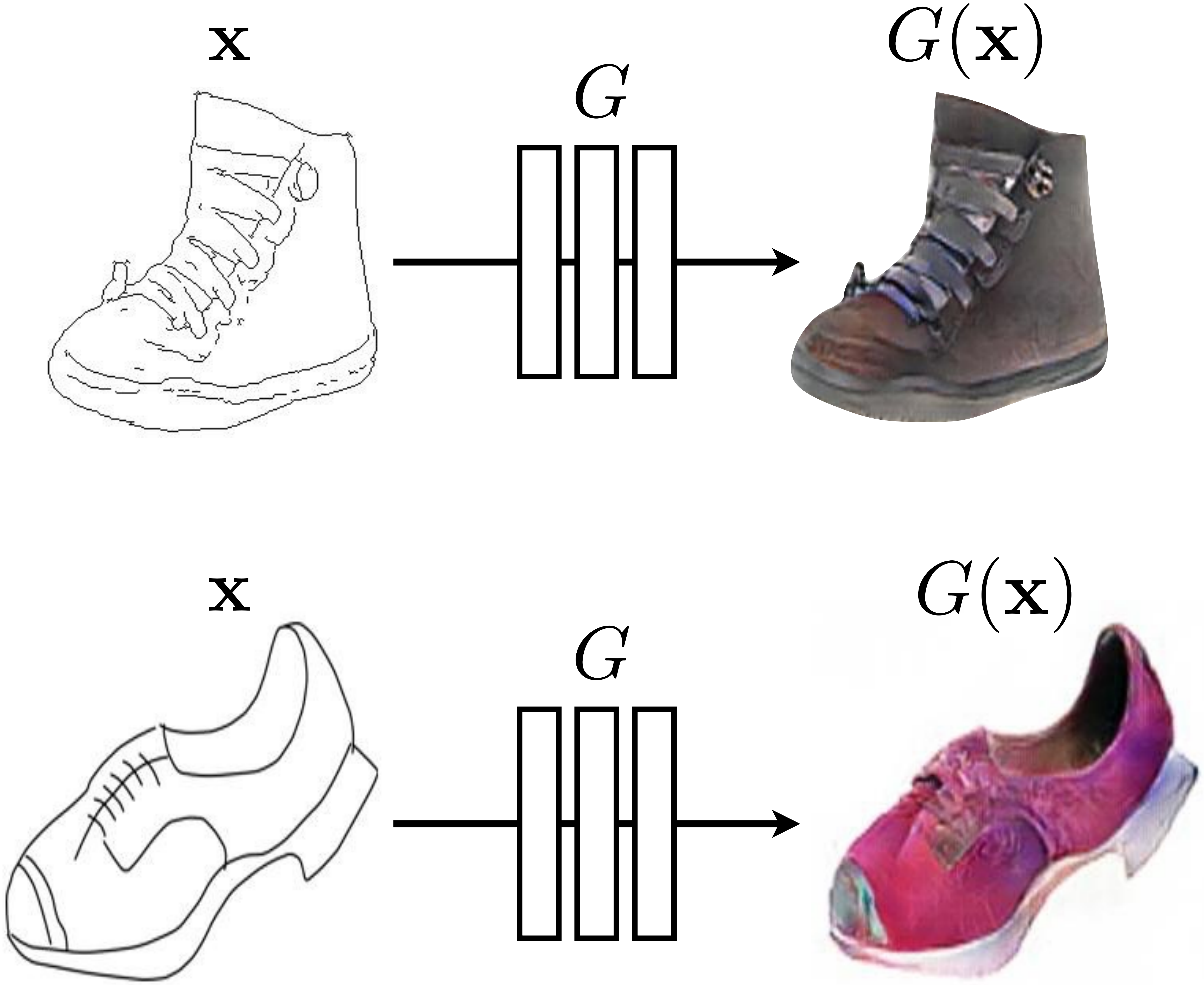


L1 loss + discriminator

Source: Isola, Freeman, Torralba

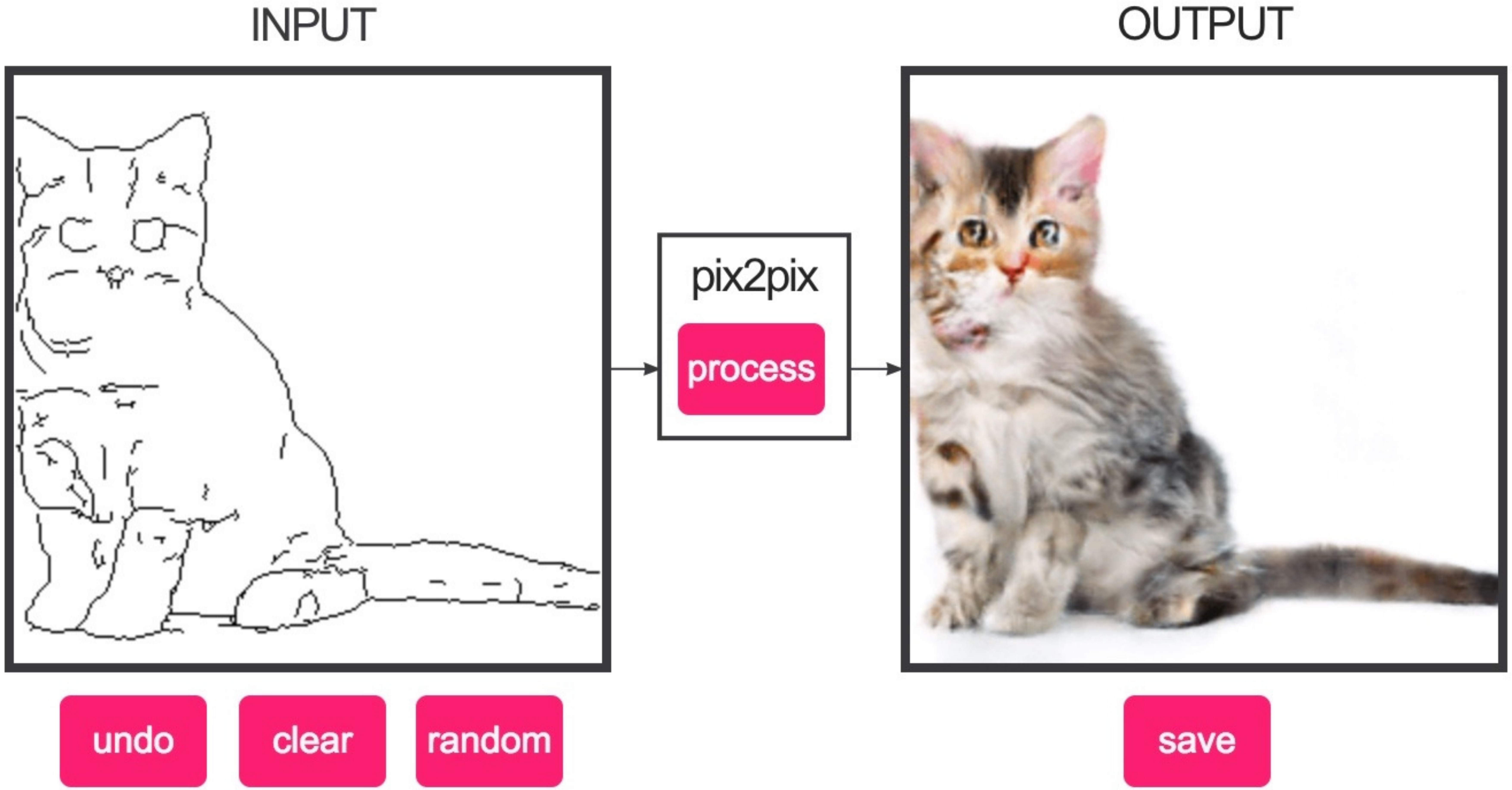
Training data

[HED, Xie & Tu, 2015]



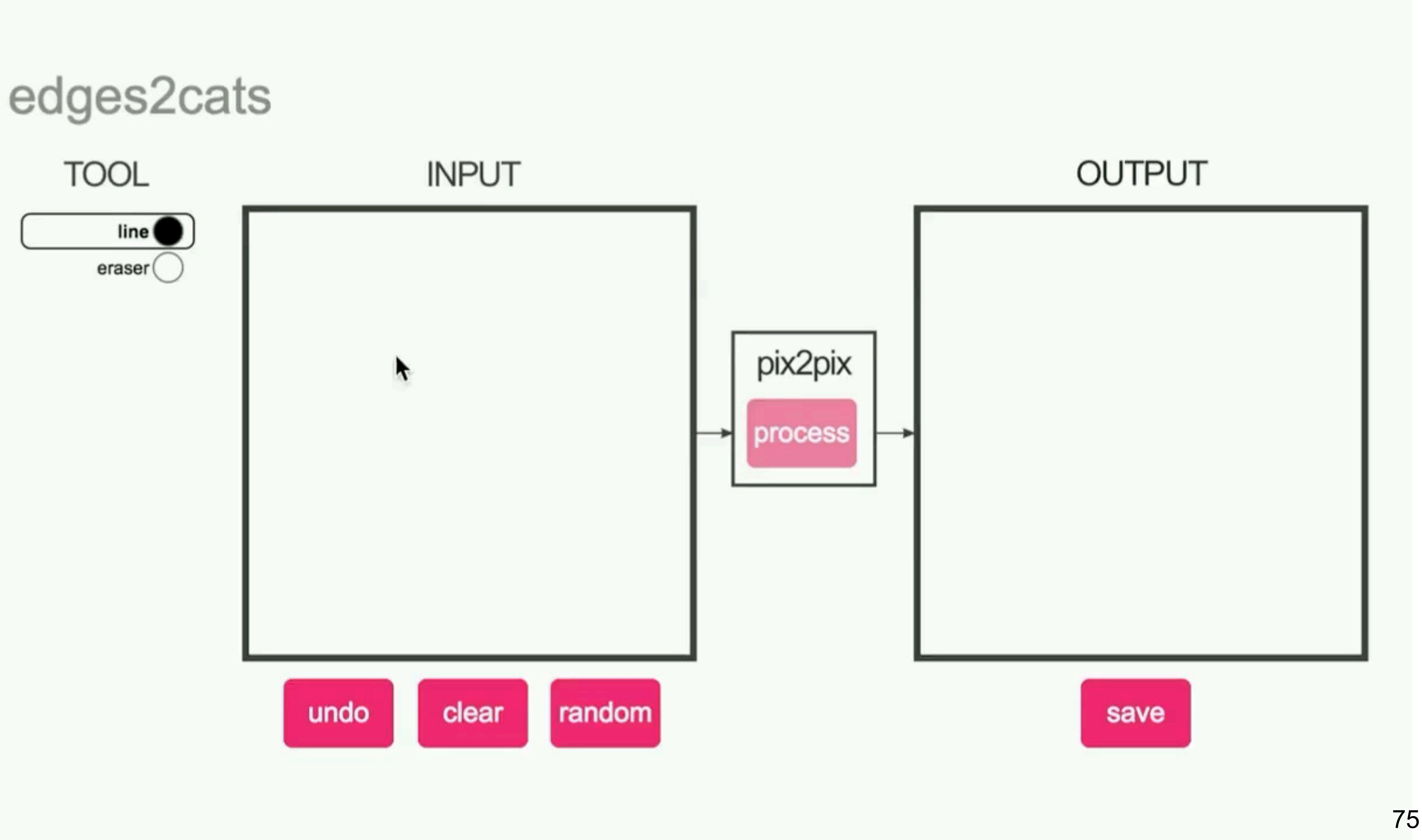
Source: Isola, Freeman, Torralba

edges2cats

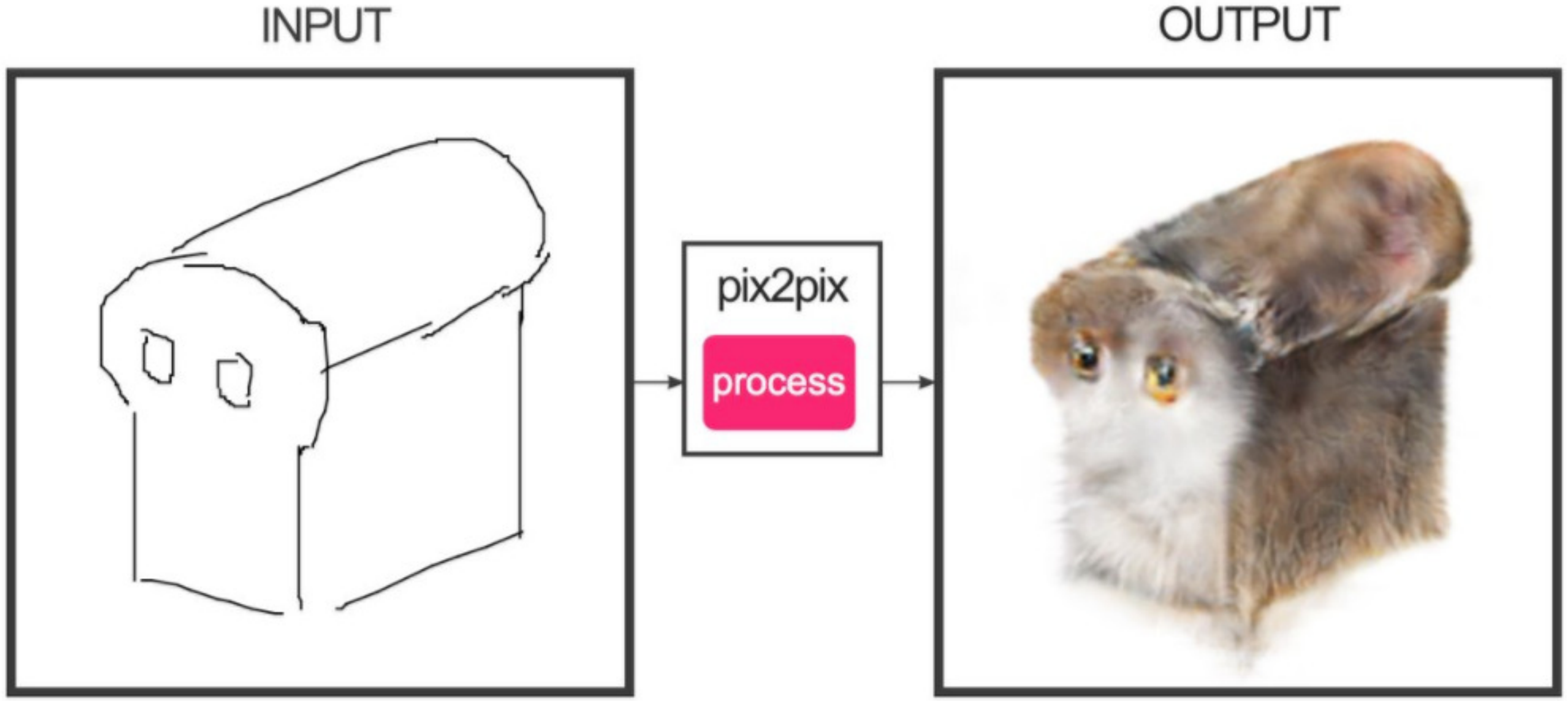


[Chris Hess, edges2cats]

Source: Isola, Freeman, Torralba



Source: Isola, Freeman, Torralba



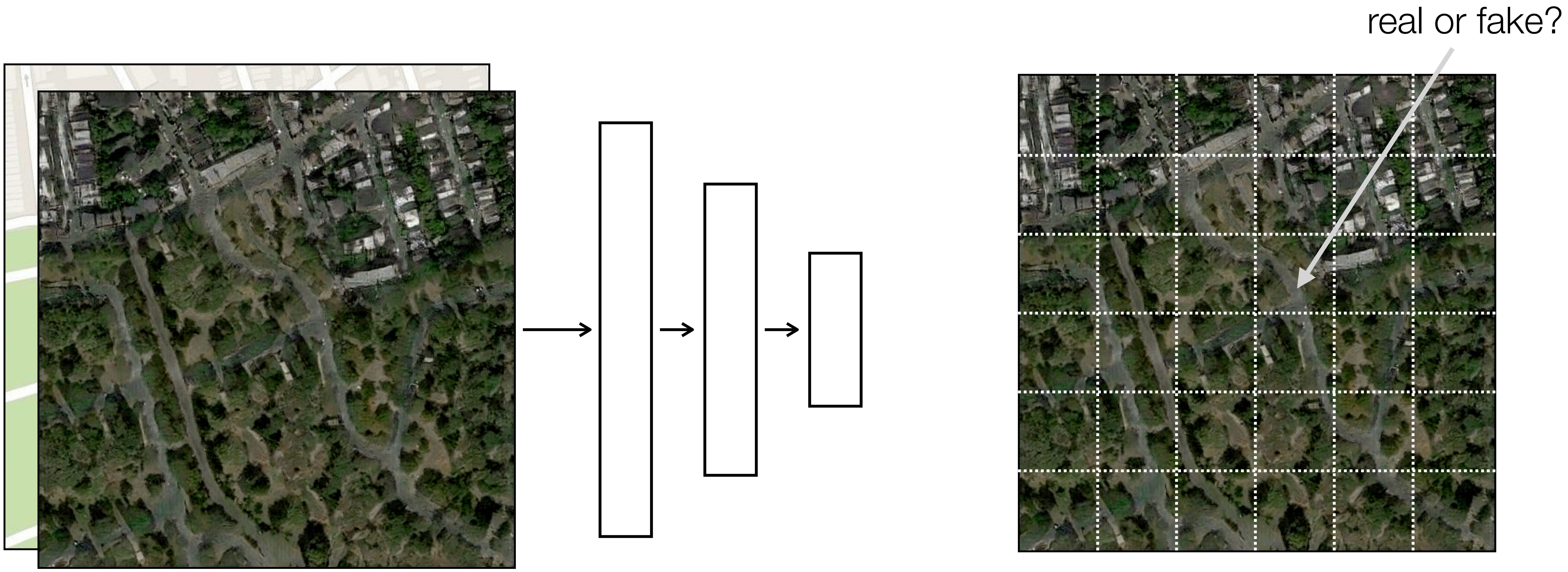
Ivy Tasi @ivymyt

Vitaly Vidmirov @vvid

Source: Isola, Freeman, Torralba

Architectures

Discriminator: fully convolutional network

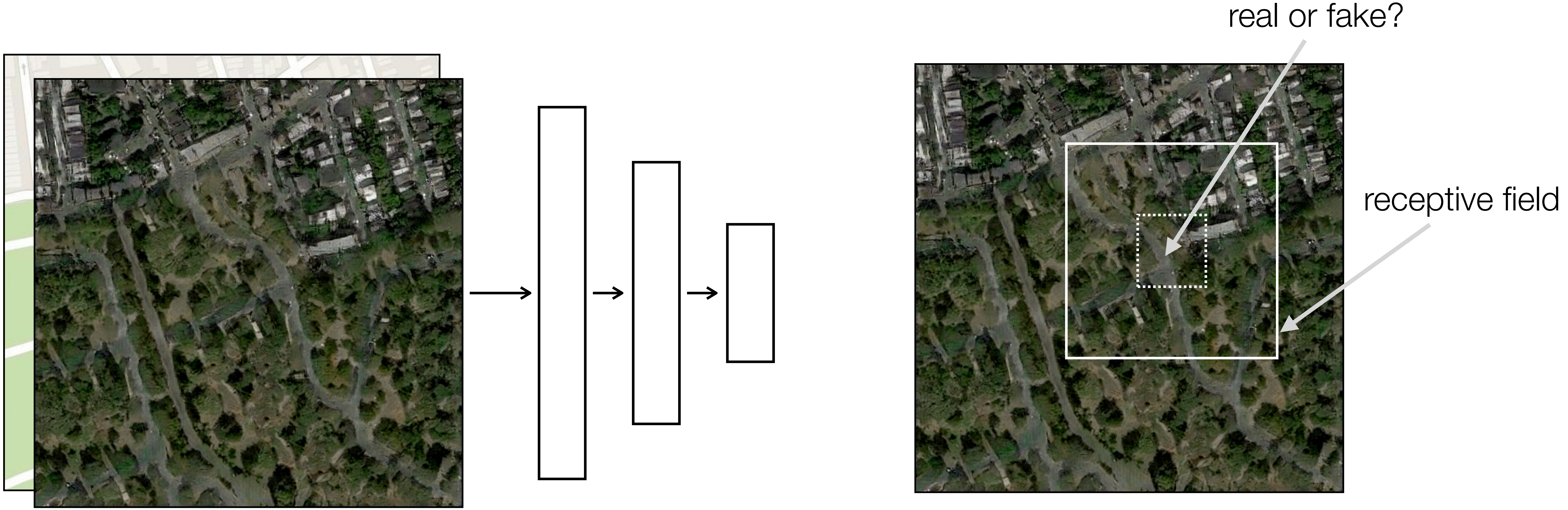


$n \times n$ output map (last conv. layer)

Sequence of strided convolutions

Architectures

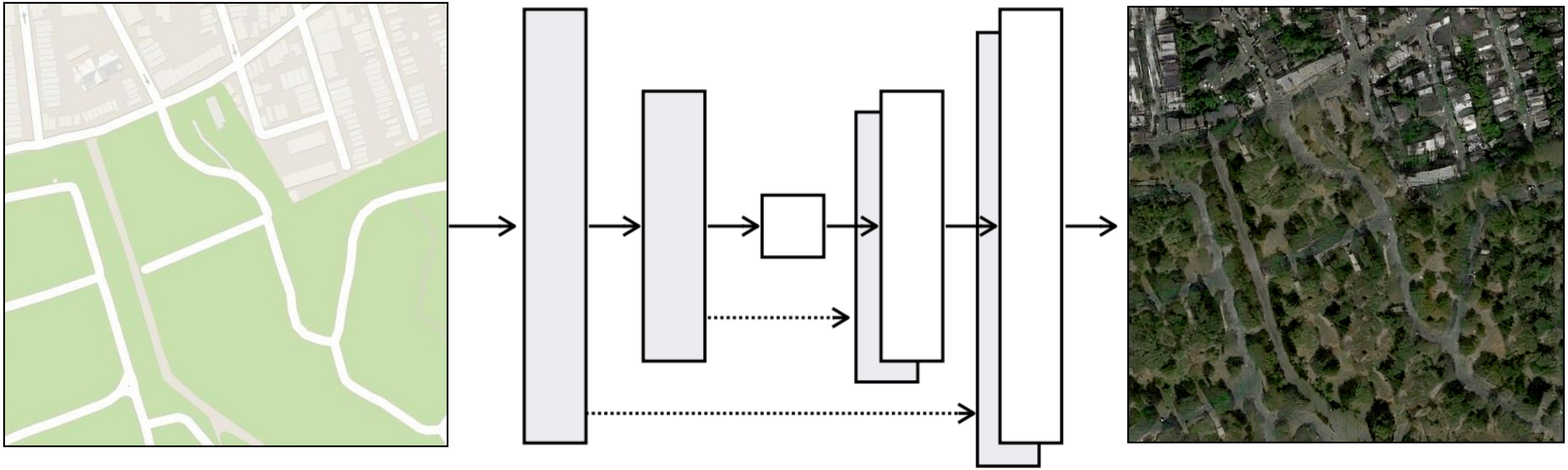
Discriminator: fully convolutional network



Also known as a **Patch GAN**, since it effectively only looks at patches

Architectures

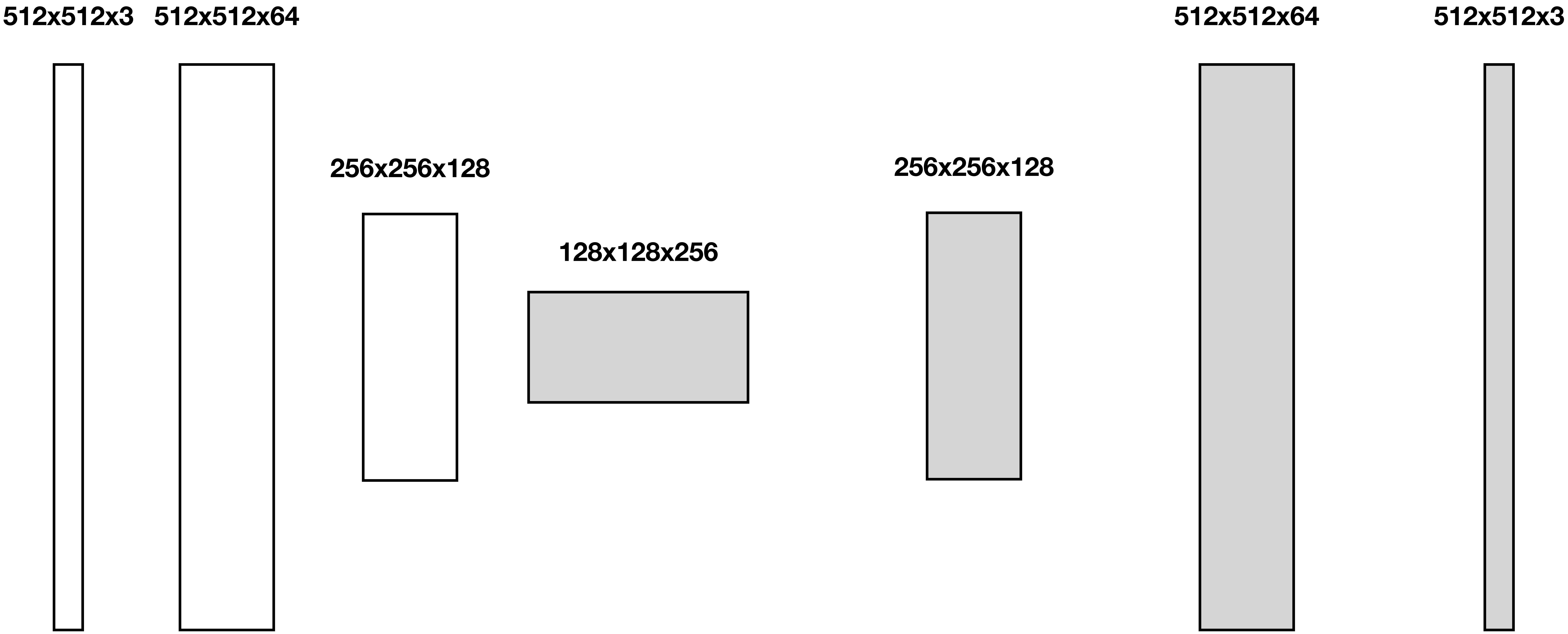
Generator: U-Net



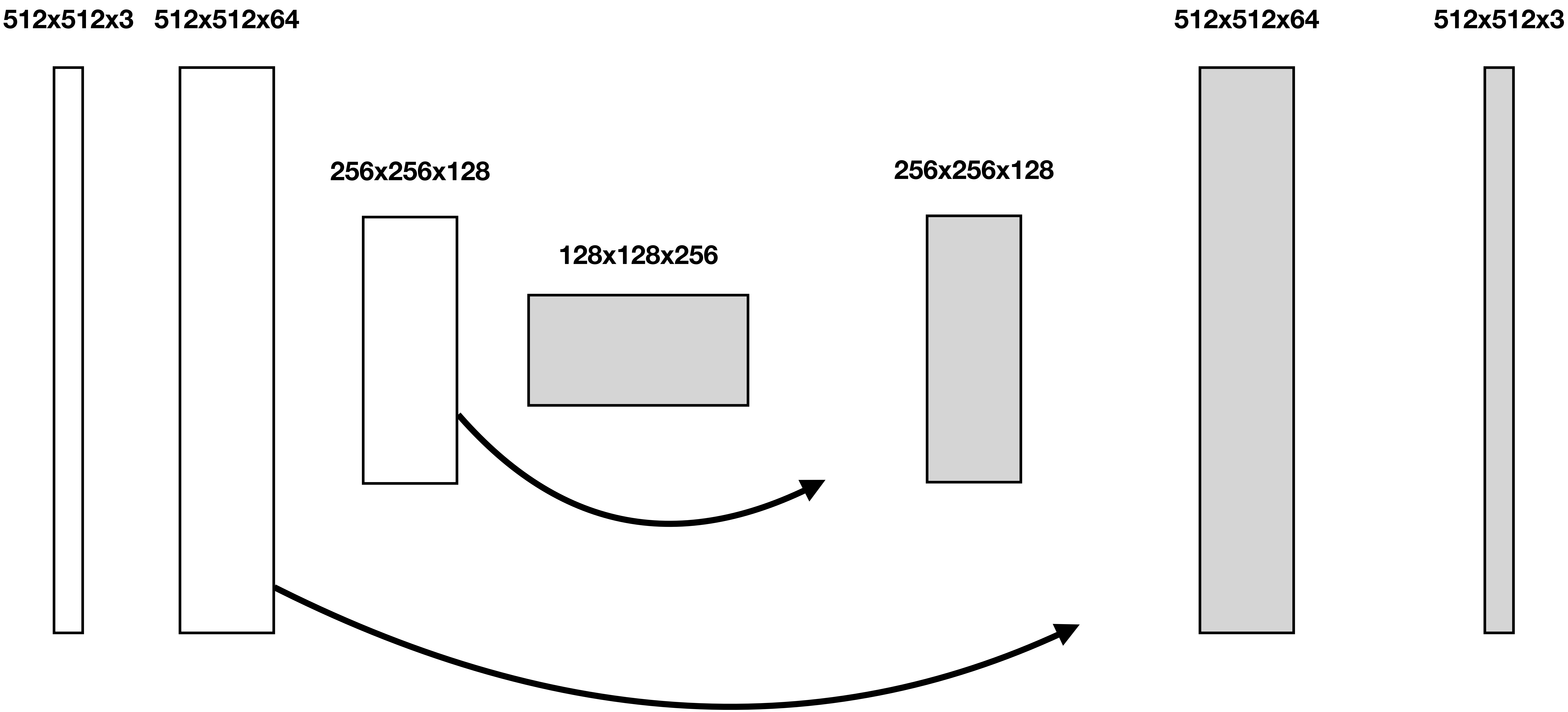
Skip connections between encoder and decoder layers

 Figure from [Isola et al., “Image-to-Image Translation with Conditional Adversarial Networks”, 2017]

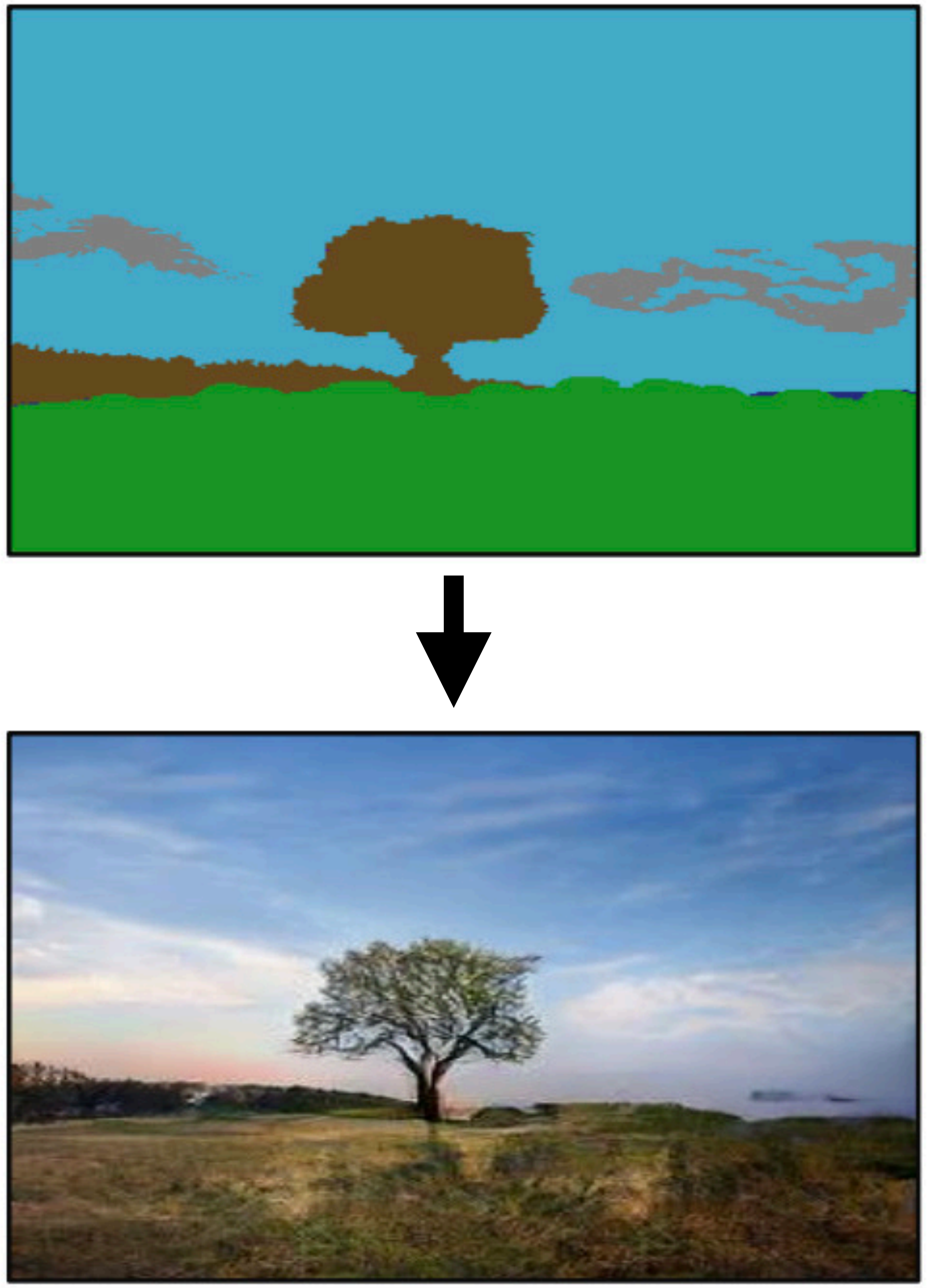
U-Net

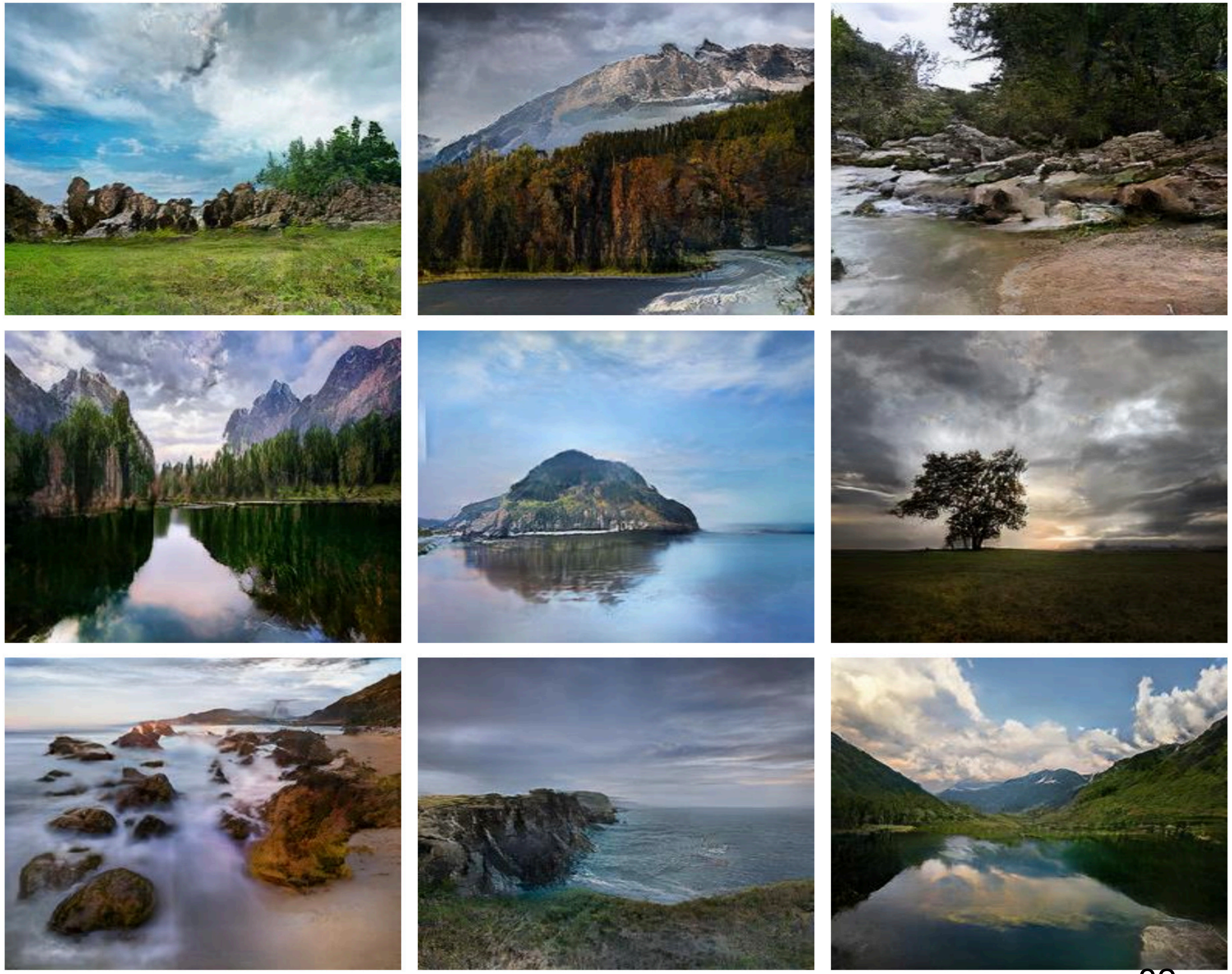

512x512x3
512x512x64
256x256x128
128x128x256
256x256x128
512x512x64
512x512x3

U-Net

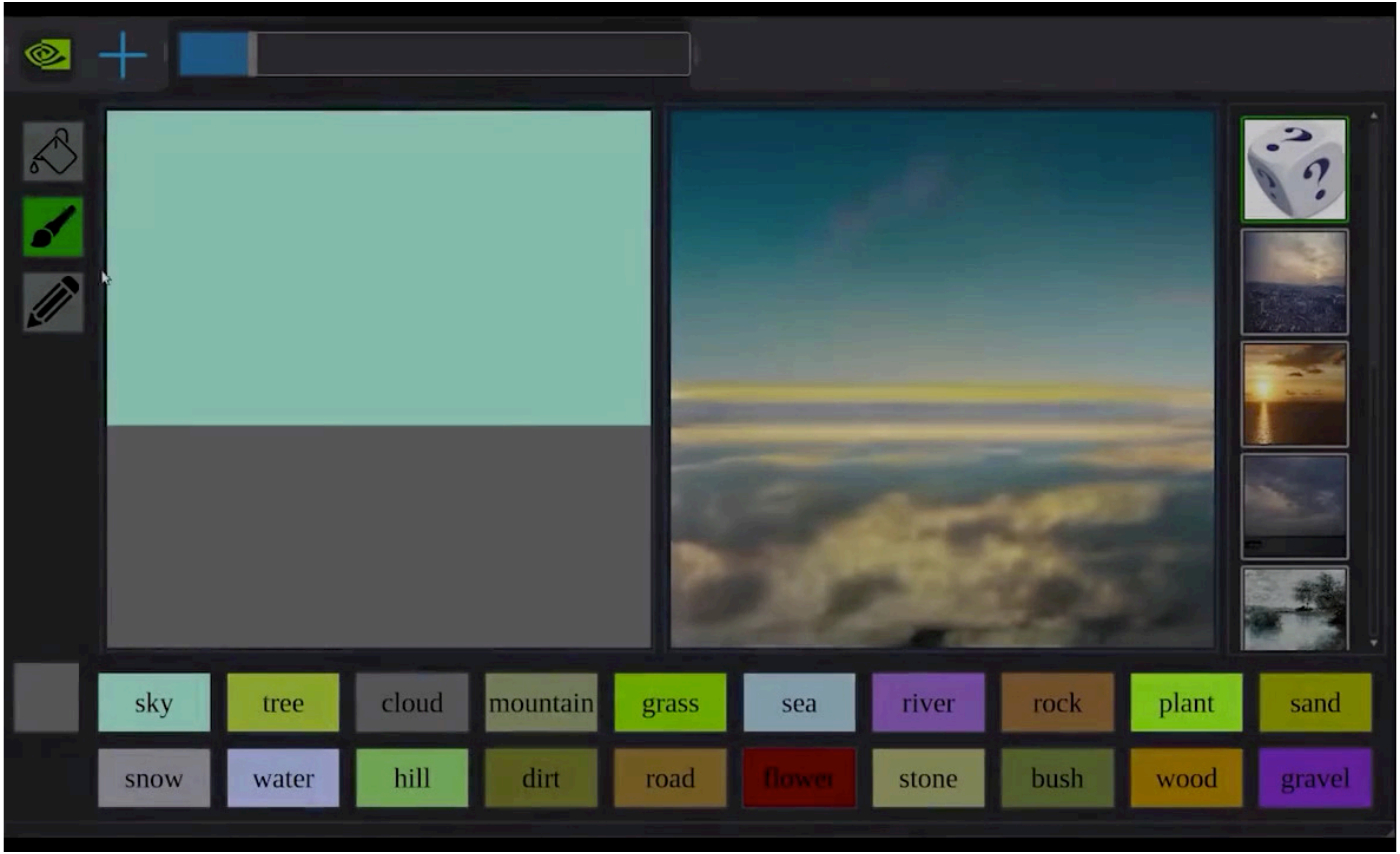

512x512x3
512x512x64
256x256x128
128x128x256
256x256x128
512x512x64
512x512x3

More recent architectures



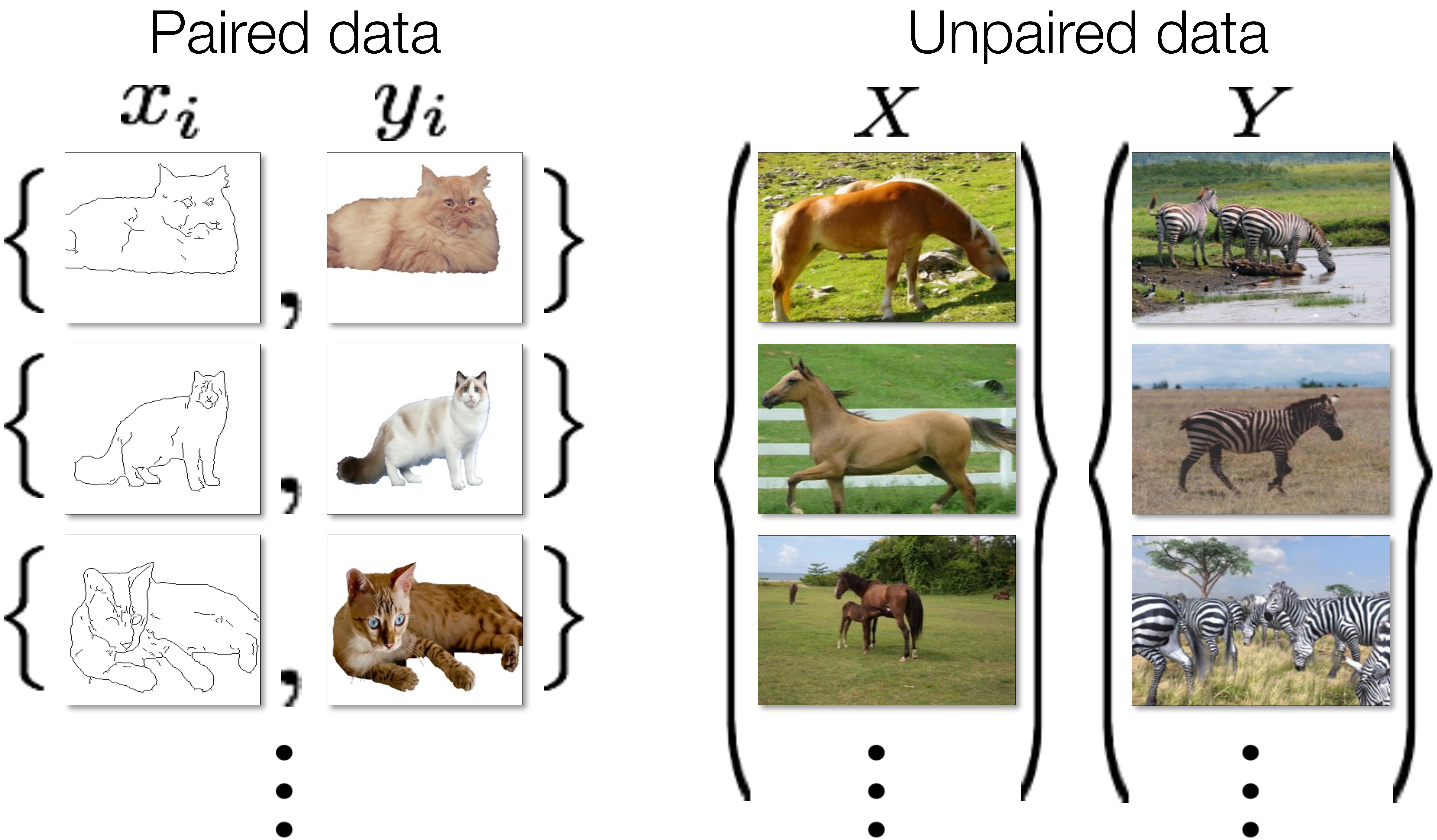


["GauGAN", Park et al., CVPR 2019]

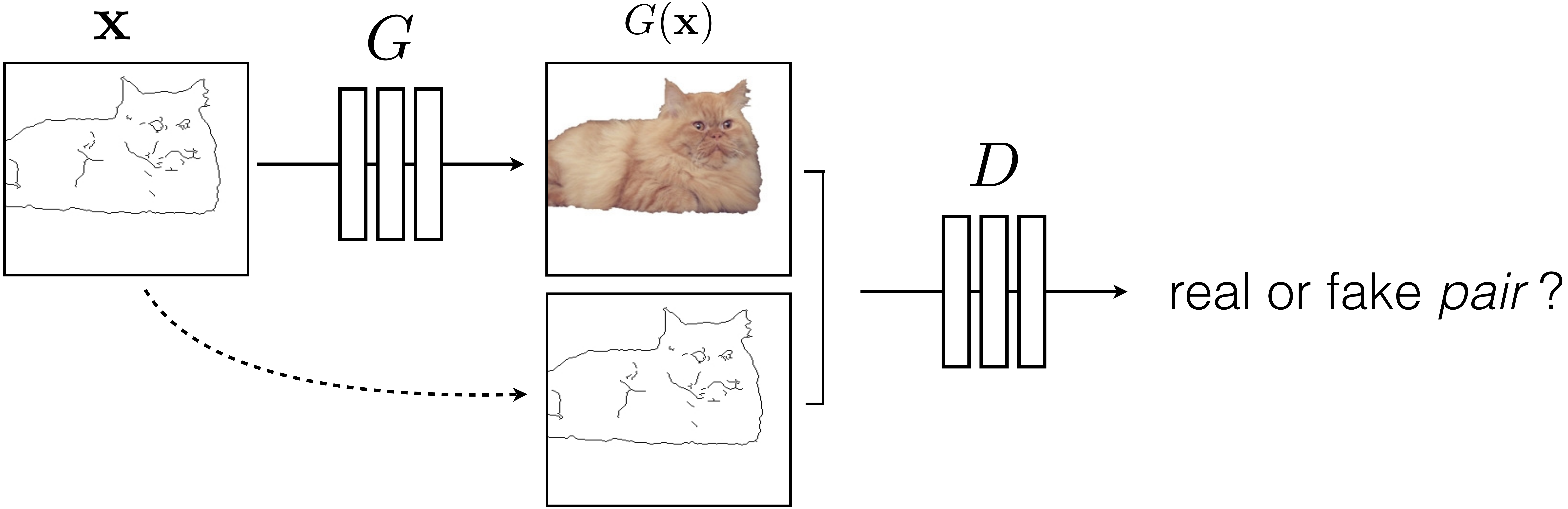


["GauGAN", Park et al., CVPR 2019]

Handling unpaired data

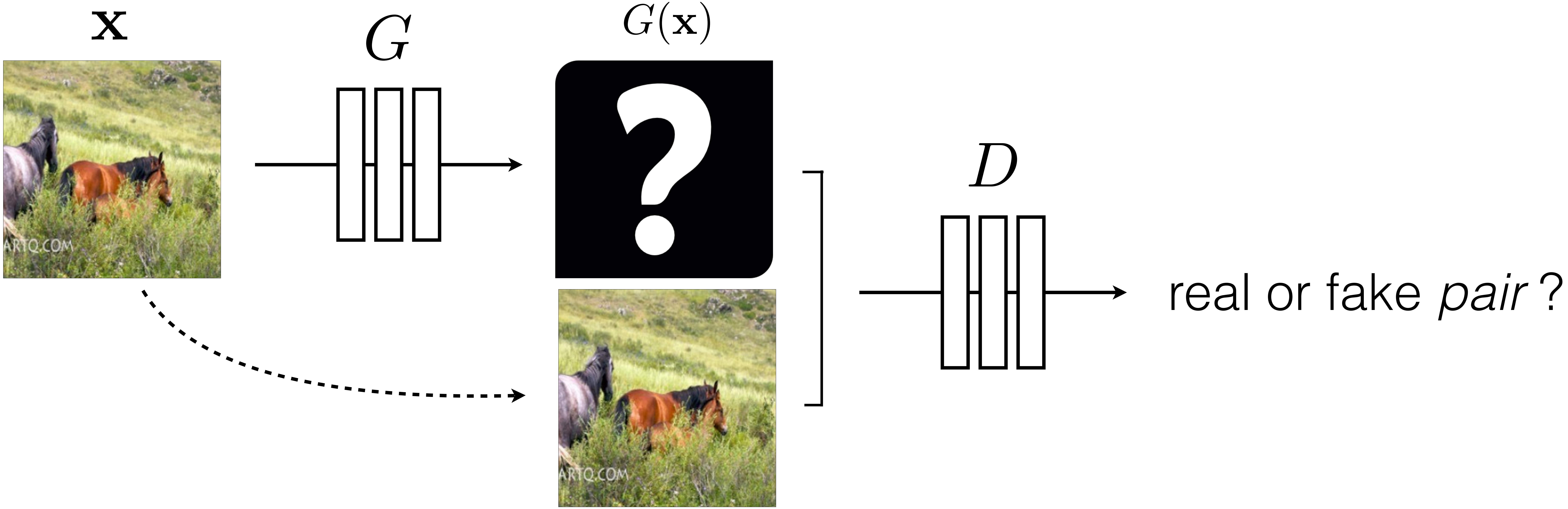


Source: Isola, Freeman, Torralba



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y})) \;]$$

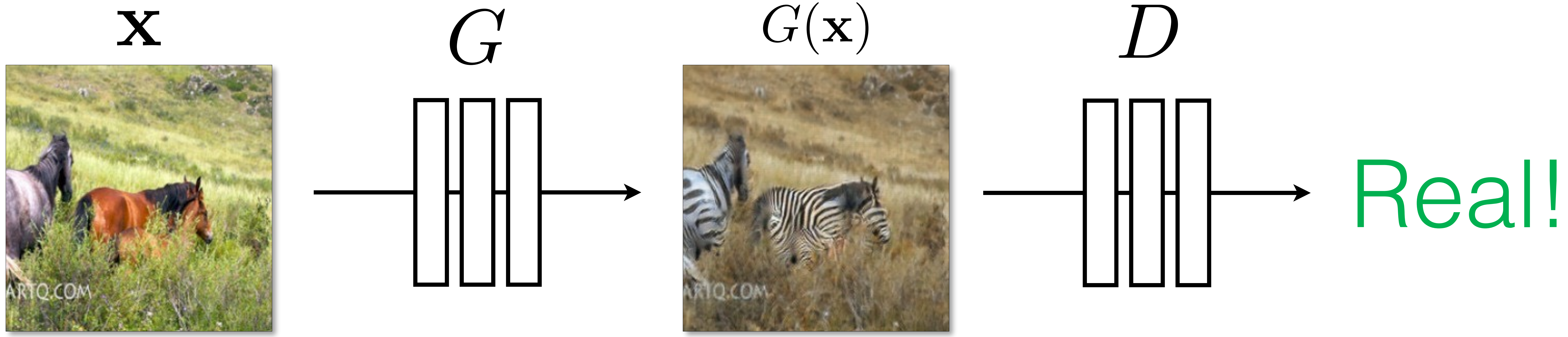
Source: Isola, Freeman, Torralba



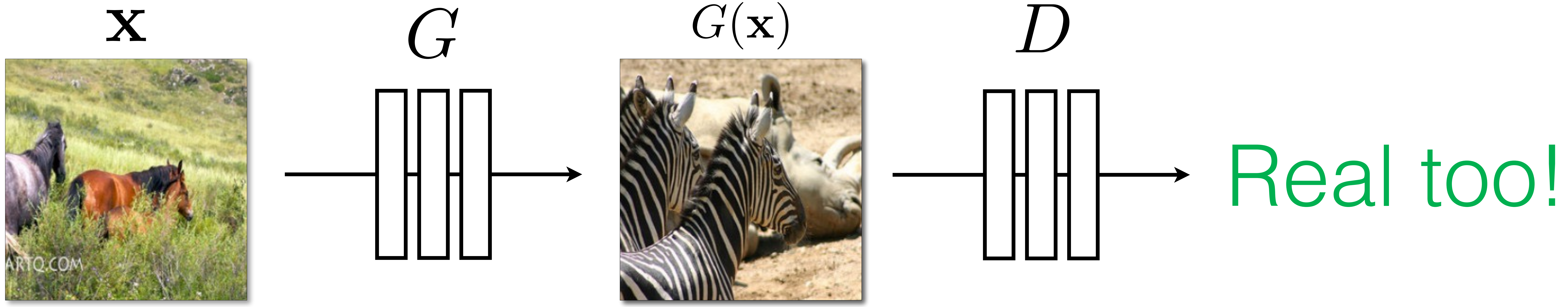
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\ ]$$

No input-output pairs!

Source: Isola, Freeman, Torralba



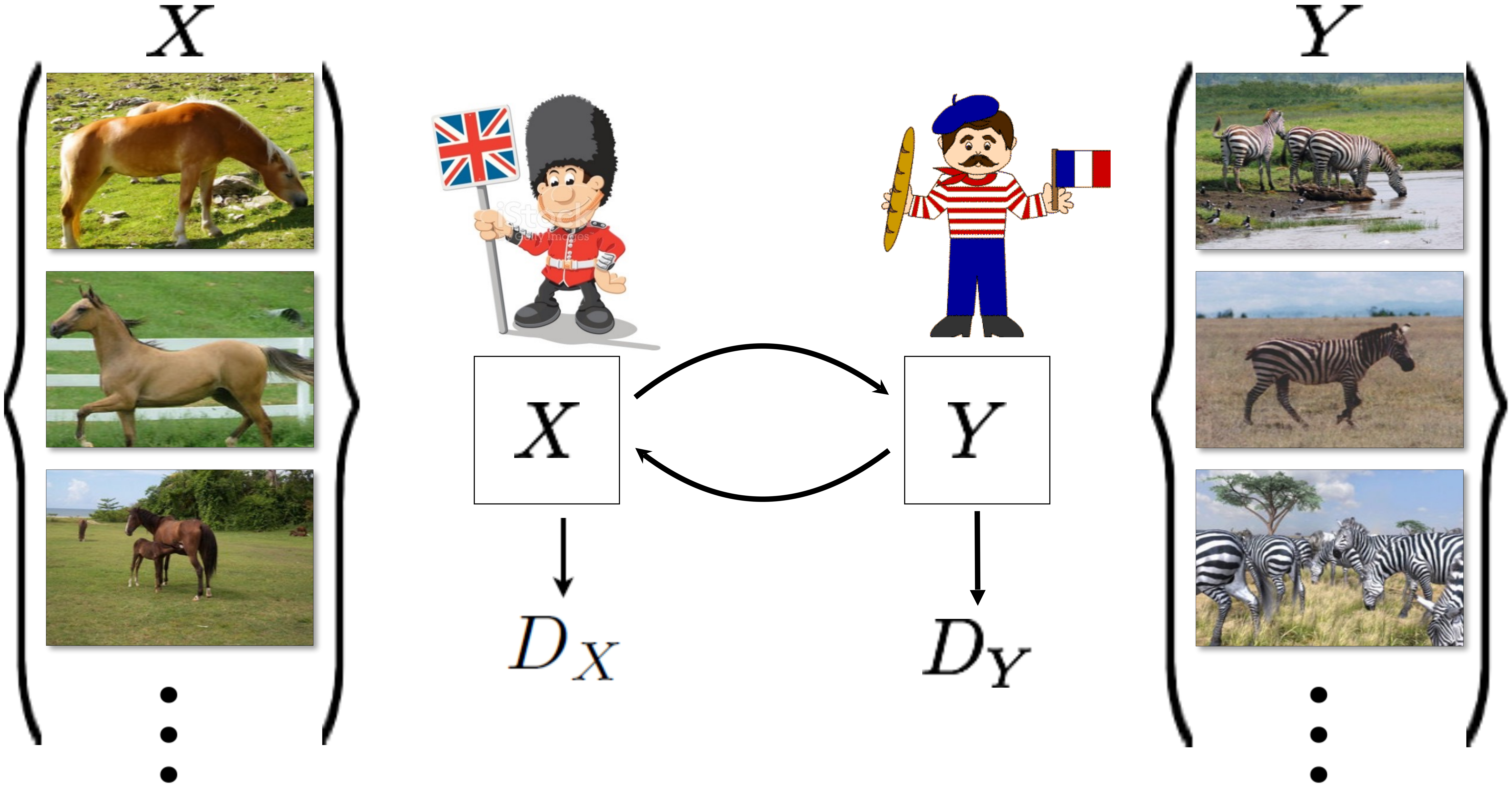
Source: Isola, Freeman, Torralba



Nothing to force output to correspond to input

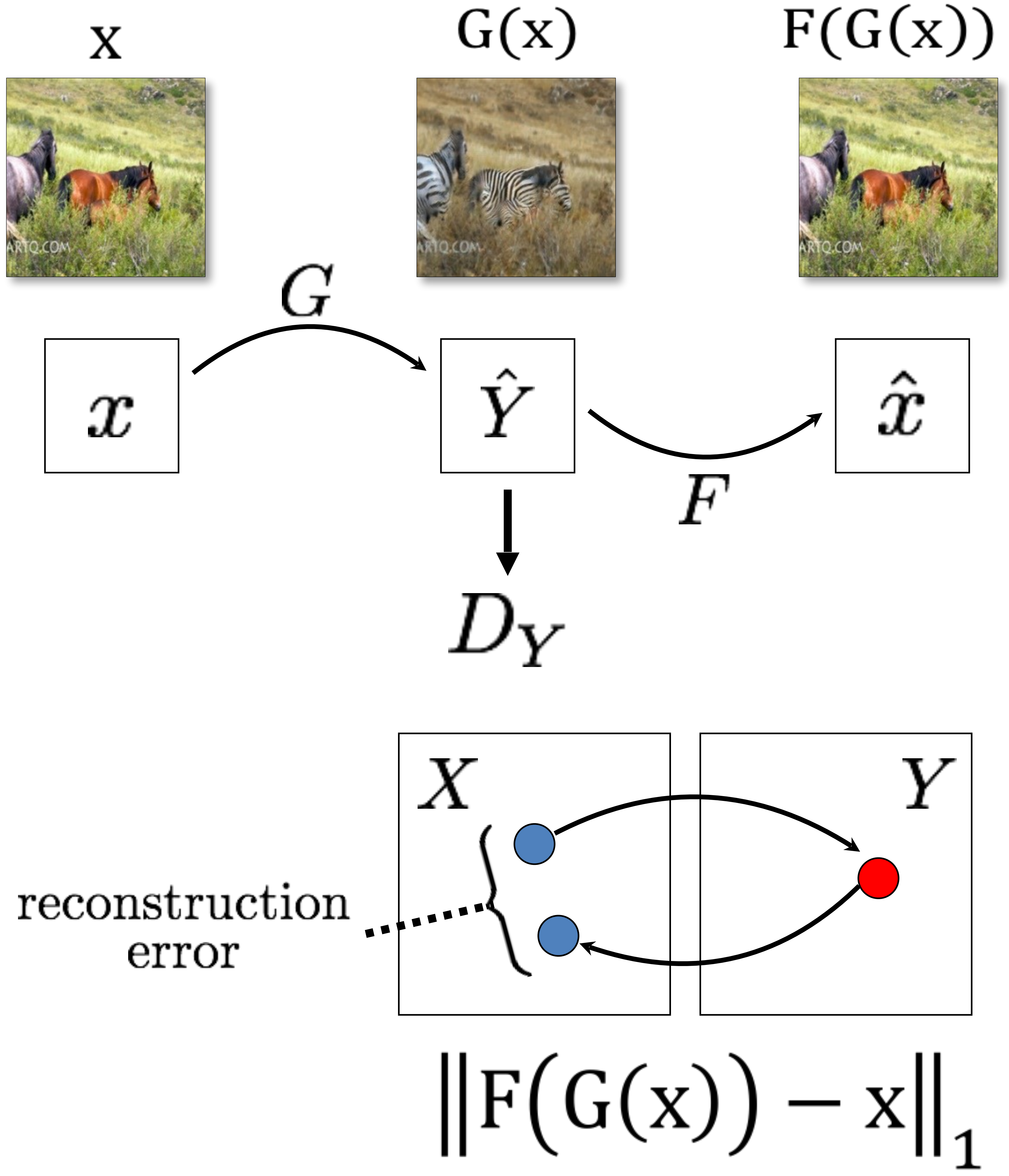
Source: Isola, Freeman, Torralba

CycleGAN



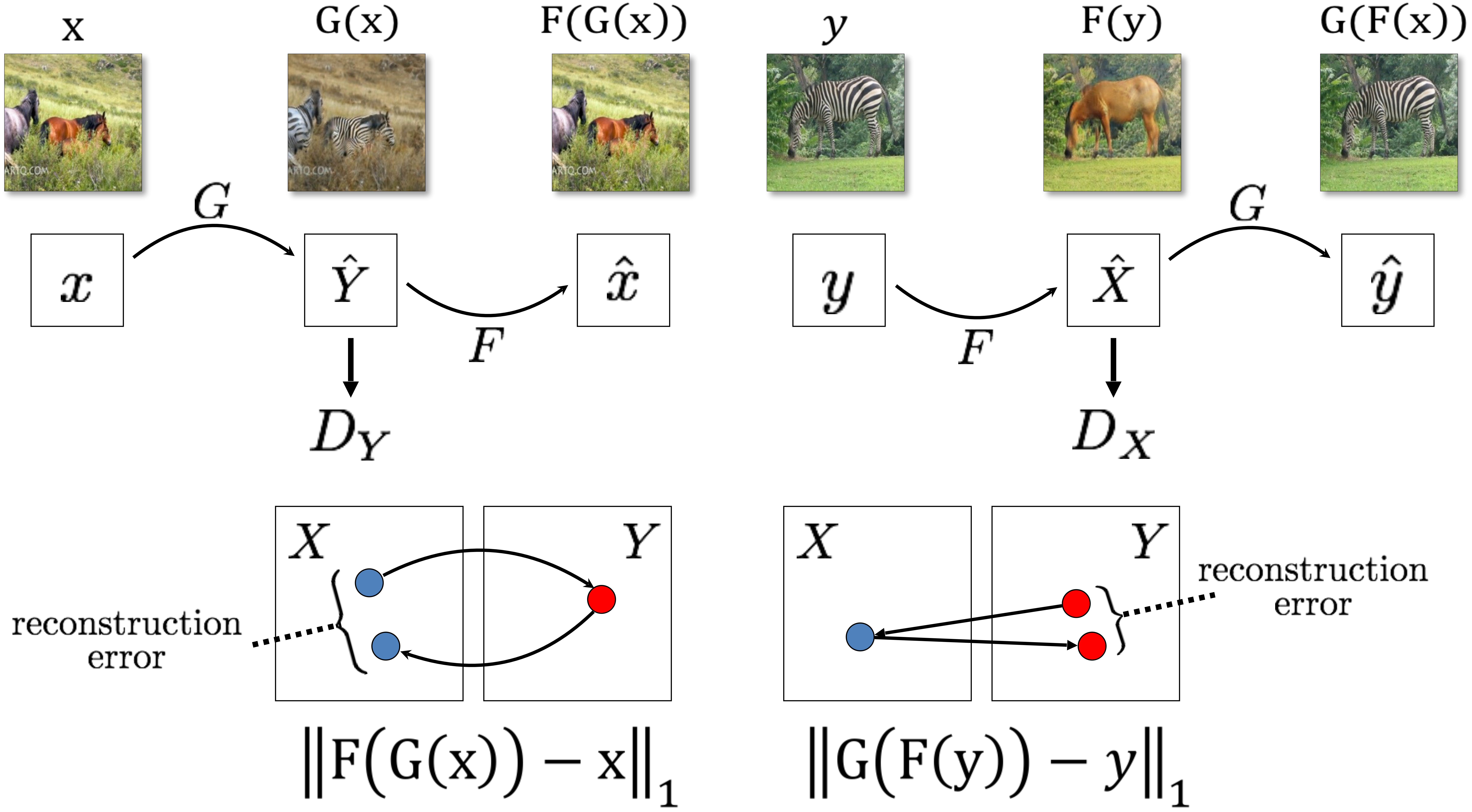
Source: Isola, Freeman, Torralba

Cycle Consistency Loss



Source: Isola, Freeman, Torralba

Cycle Consistency Loss



Source: Isola, Freeman, Torralba

Source: Isola, Freeman, Torralba

Source: Isola, Freeman, Torralba


Input
Monet
Van Gogh
Cezanne
Ukiyo-e

Next time: more image synthesis