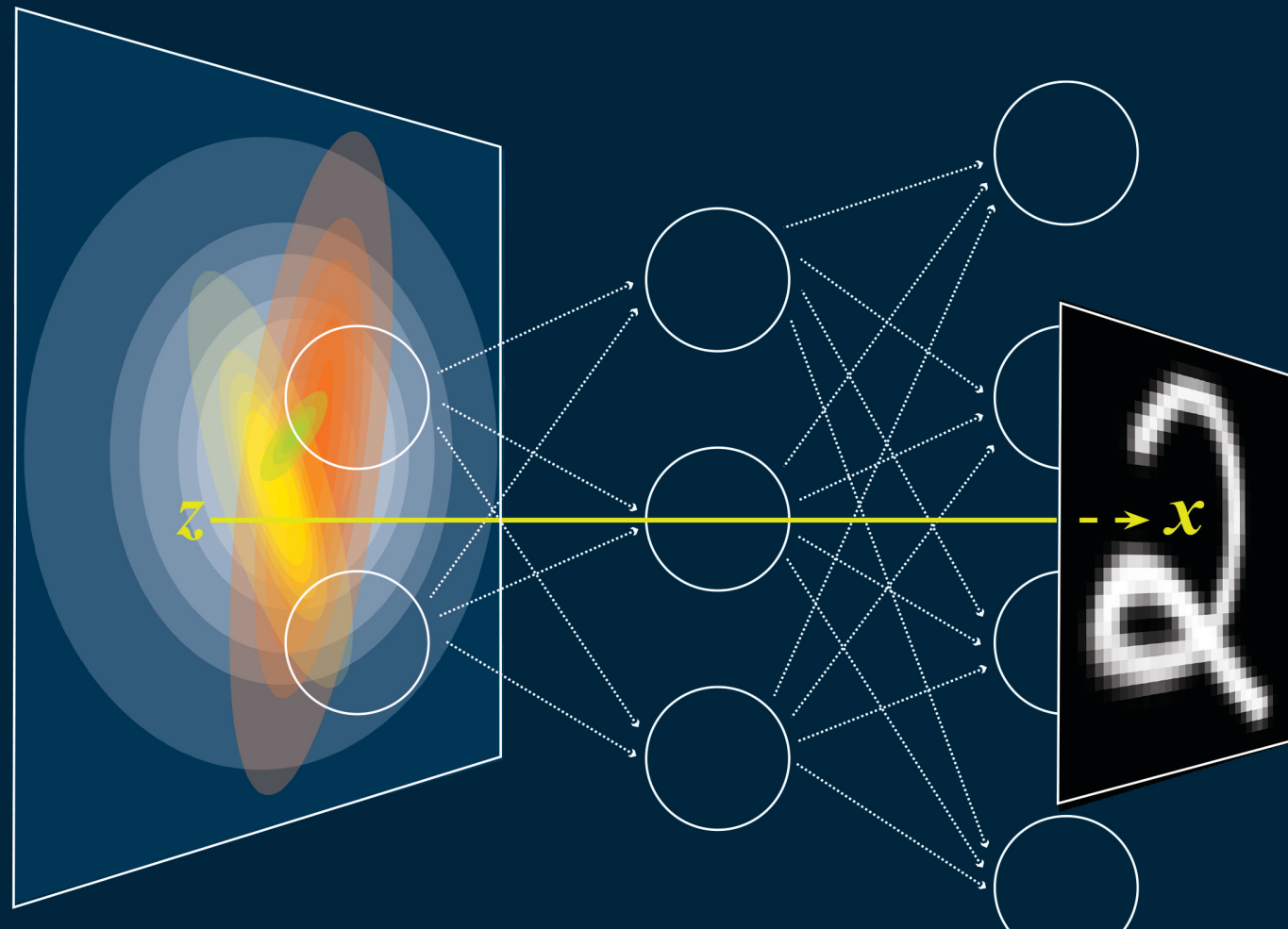




Style and Content, Texture Synthesis

Jun-Yan Zhu

16-726, Spring 2022



Probabilistic Machine Learning

Advanced Topics

Kevin P. Murphy

IV	Generation	767	
21	Generative models: an overview		769
22	Variational autoencoders	783	
23	Auto-regressive models	827	
24	Normalizing Flows	835	
25	Energy-based models	855	
26	Denoising diffusion models	875	
27	Generative adversarial networks		883

Collection Style Transfer



Photograph ©Alexei Efros



Monet



Van Gogh



Cezanne



Ukiyo-e

Style and Content Separation

A

Classification

A	B	C	D	E
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
A	B	C	D	E
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
A	B	C	D	E
B	C	A	E	D

Domain Adaptation

B

Extrapolation

A	B	C	D	E
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
A	B	C	D	E
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
A	B	C	D	E
?	?	C	D	E

Paired Image-to-Image Translation

C

Translation

A	B	C	D	E	?	?	?
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>			
A	B	C	D	E			
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>			
A	B	C	D	E	?	?	?
?	—	—	—	?	F	G	H

Unpaired Image-to-Image Translation

Training
Generalization

Separating Style and Content
[Tenenbaum and Freeman 1996]

$$y_k^{sc} = \sum_{i=1}^I \sum_{j=1}^J w_{ijk} a_i^s b_j^c.$$

Style and Content

Adversarial loss

$$\mathbb{E}_x \log(1 - D_Y(G(x))) + \mathbb{E}_y \log D_Y(y)$$



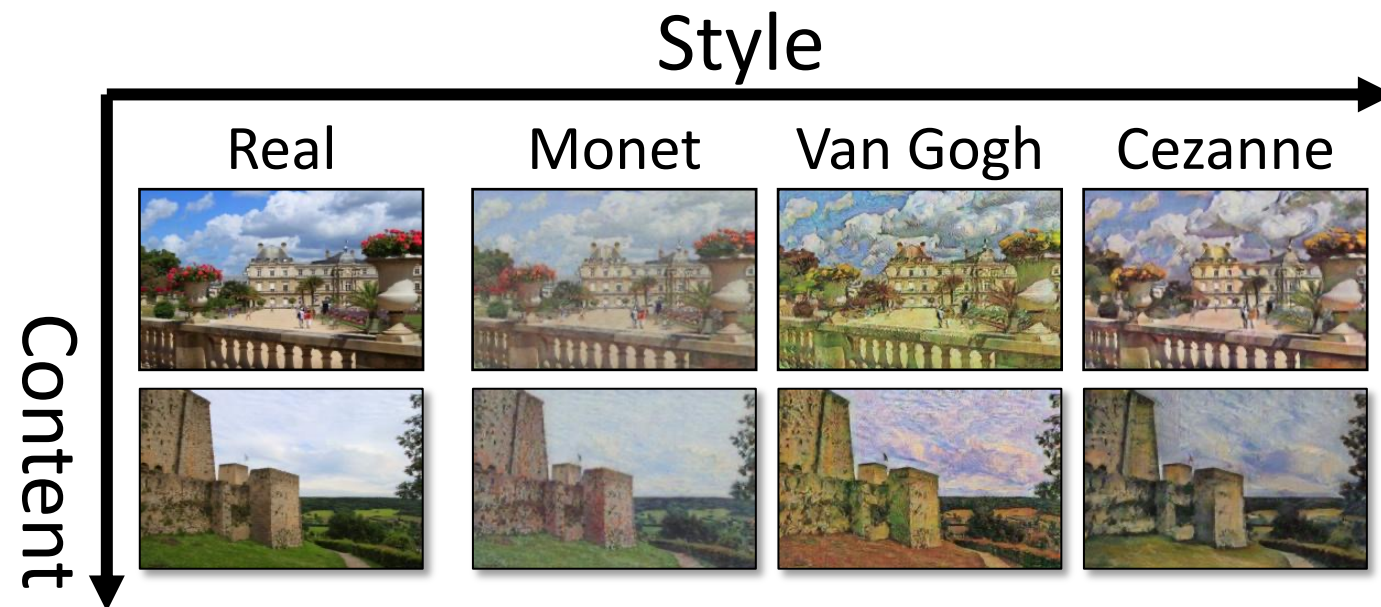
$p(x) \rightarrow p(y)$ change **style**

Cycle-consistency loss

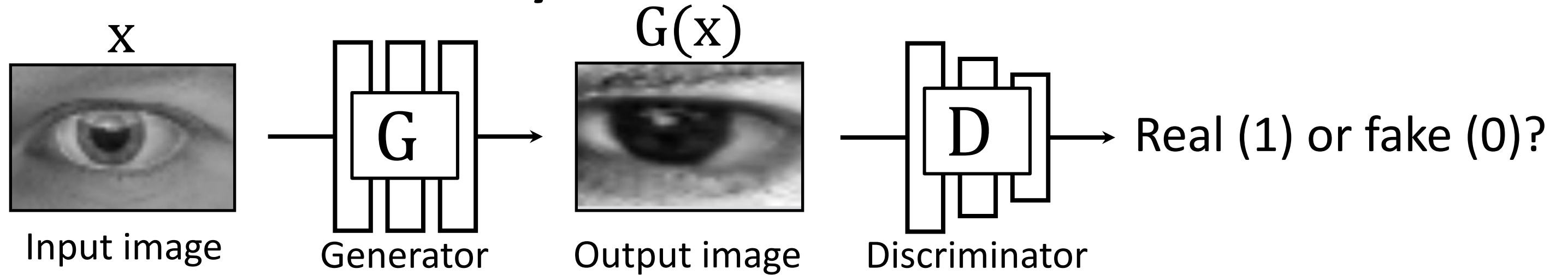
$$\mathbb{E}_x ||F(G(x)) - x||_1$$



Bidirectional: preserve **content**



Style and Content

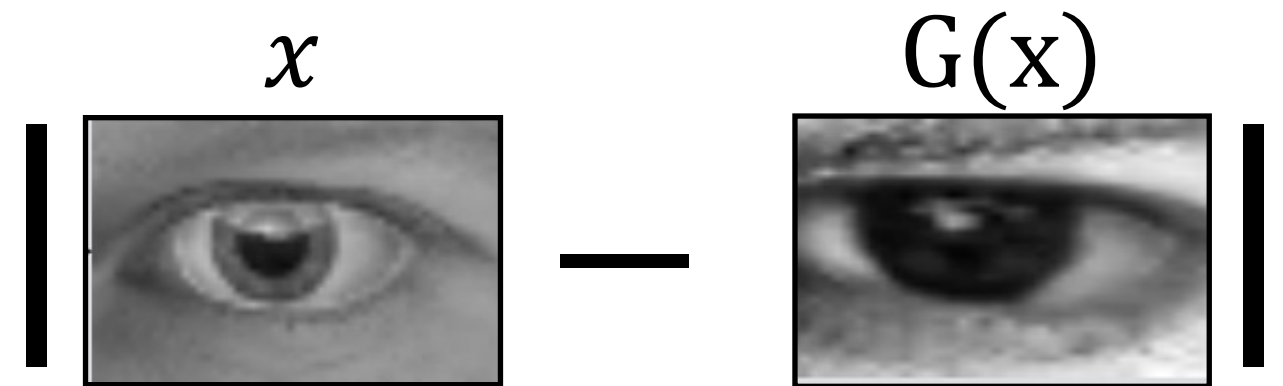


Adversarial loss (change style)

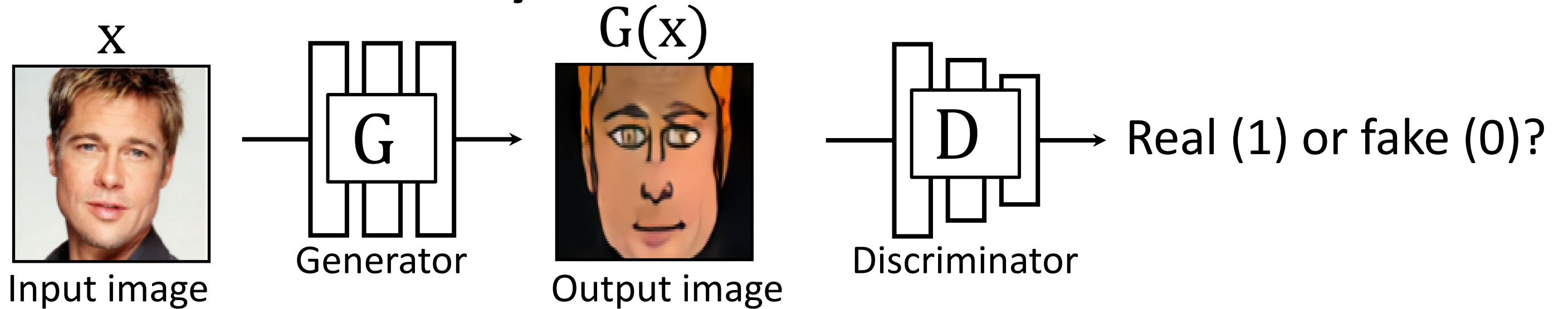
$$\mathbb{E}_x \log(1 - D_Y(G(x))) + \mathbb{E}_y \log D_Y(y)$$

L1 loss (preserve content in pixel space)

$$\mathbb{E}_x ||G(x) - x||_1$$



Style and Content

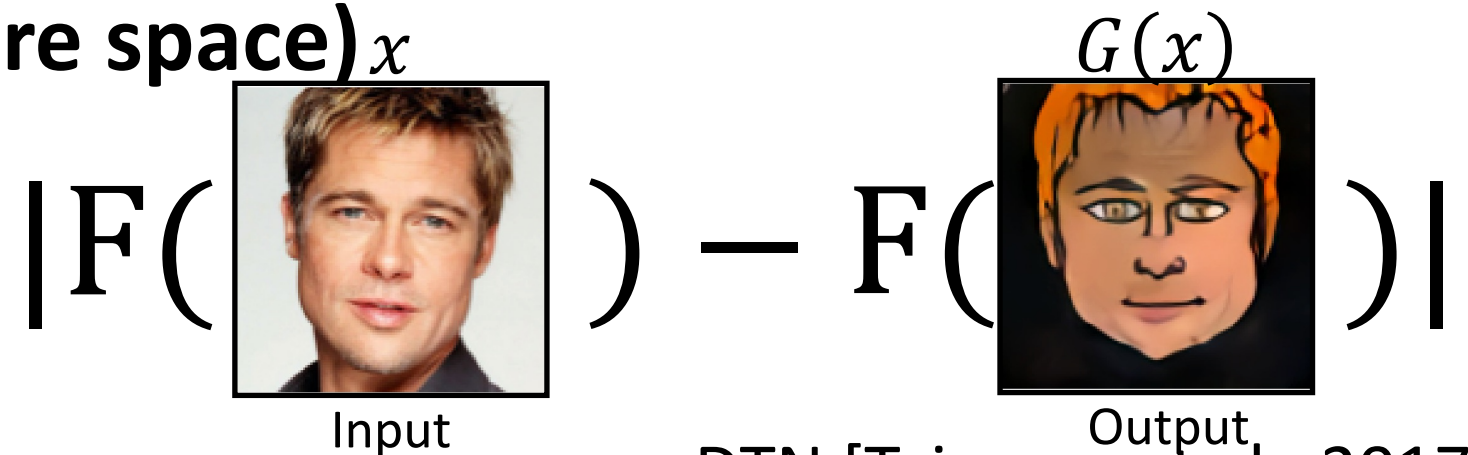


Adversarial loss (change style)

$$\mathbb{E}_x \log(1 - D_Y(G(x))) + \mathbb{E}_y \log D_Y(y)$$

Feature loss (Preserve content in feature space) _{x}

$$\mathbb{E}_x ||F(G(x)) - F(x)||$$



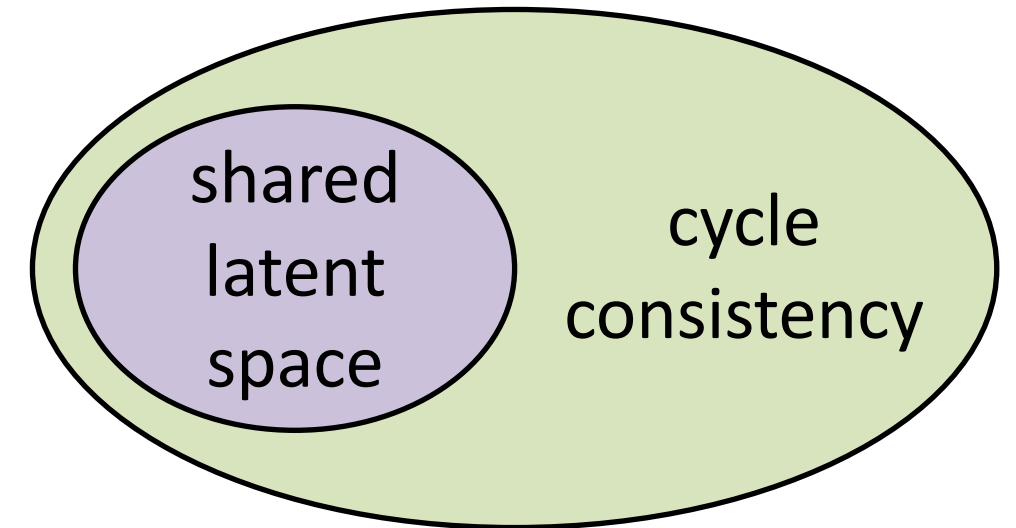
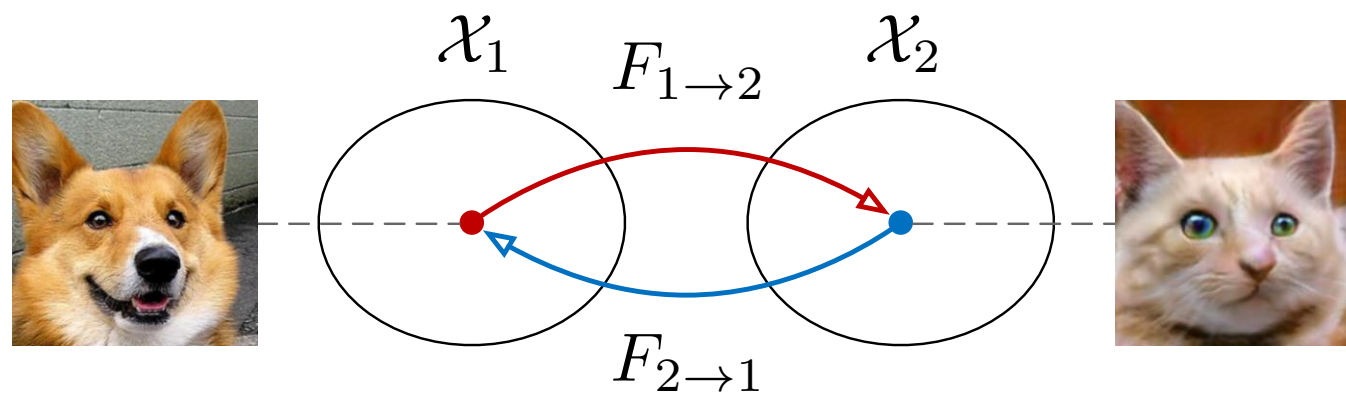
DTN [Taigman et al., 2017]

Style and Content

- Style: domain-specific features
(horse vs. zebra)
- Content: features shared across two domains

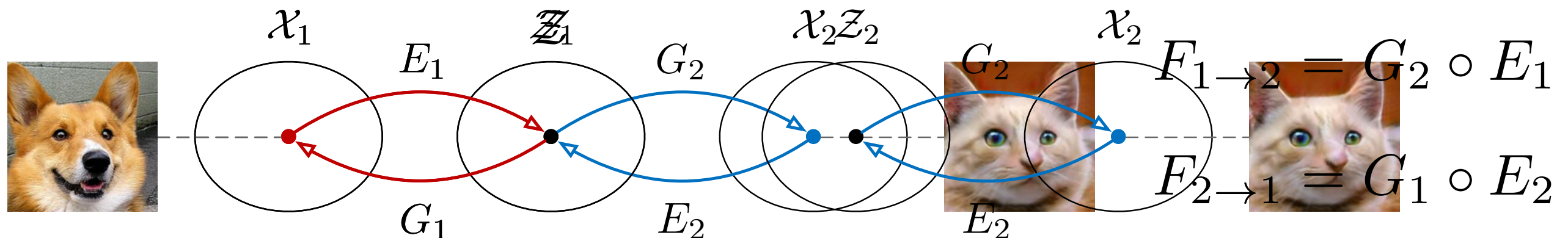
CycleGAN and UNIT

- CycleGAN (cycle consistency)



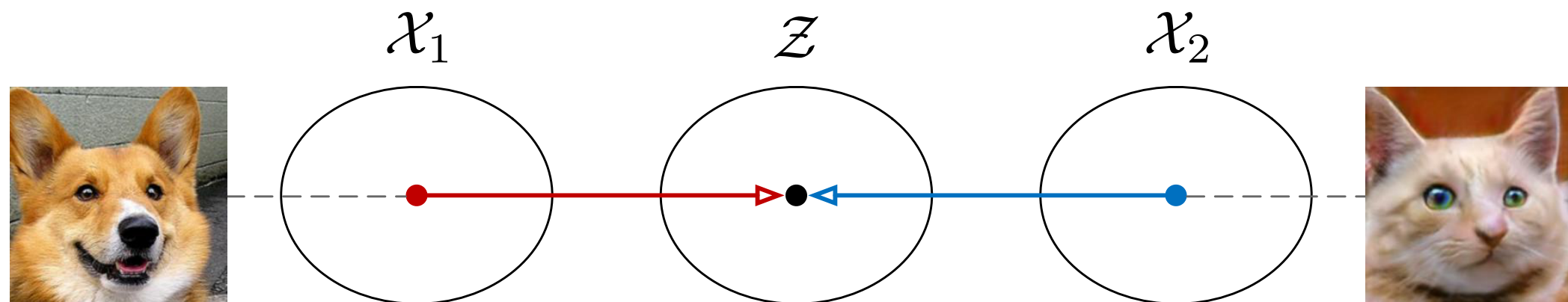
- UNIT (shared latent space) [Liu et al. 2017]

shared latent space \Rightarrow cycle consistency



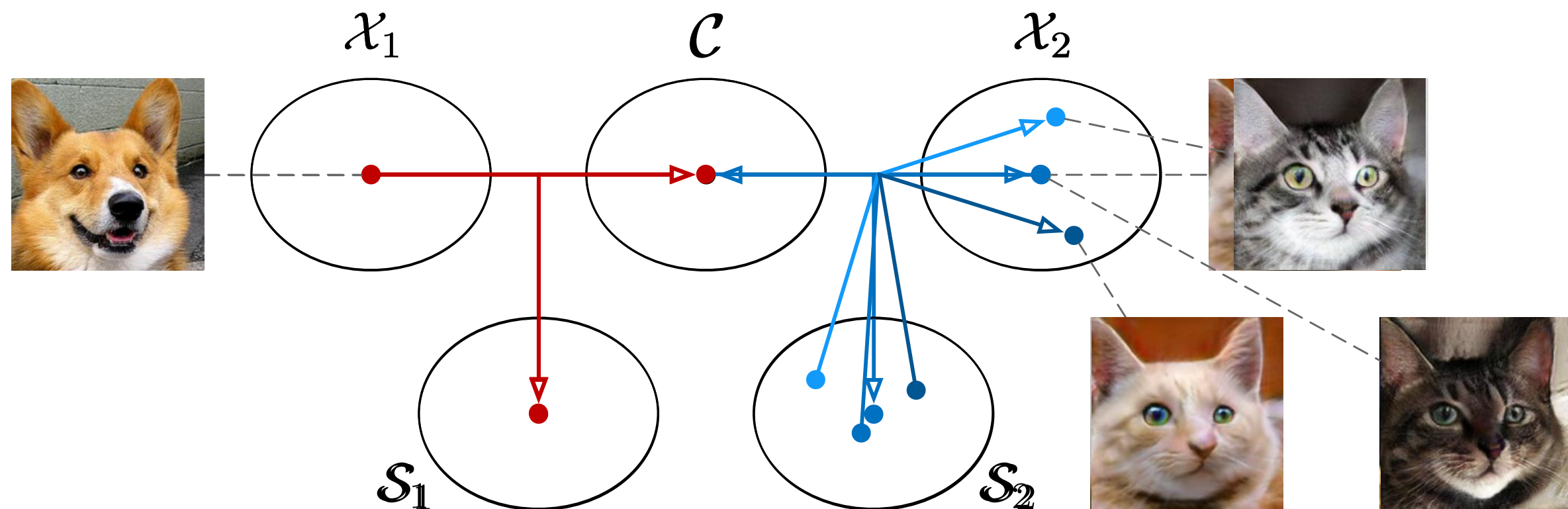
Disentangling the Latent Space

- UNIT
 - A single **shared, domain-invariant** latent space \mathcal{Z}



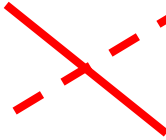
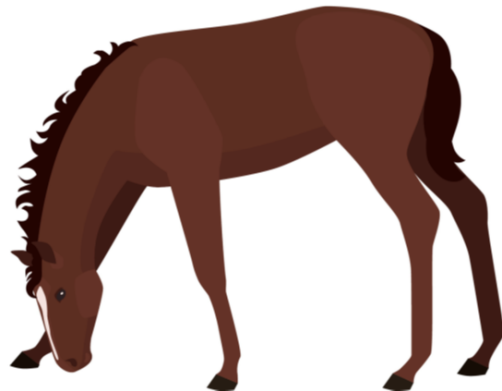
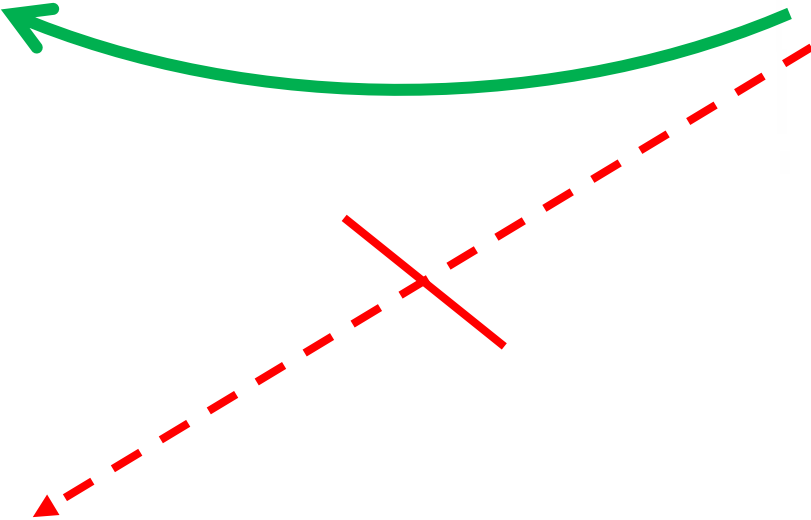
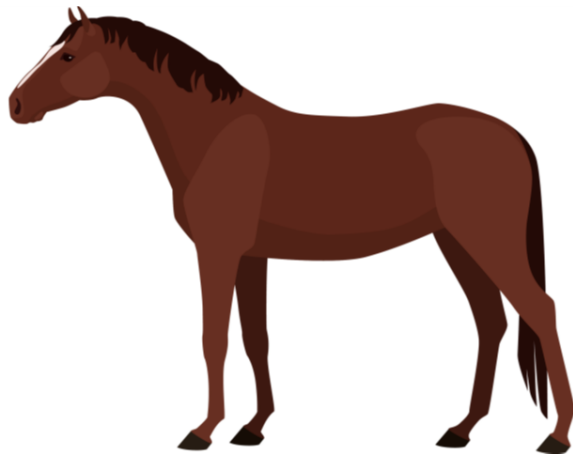
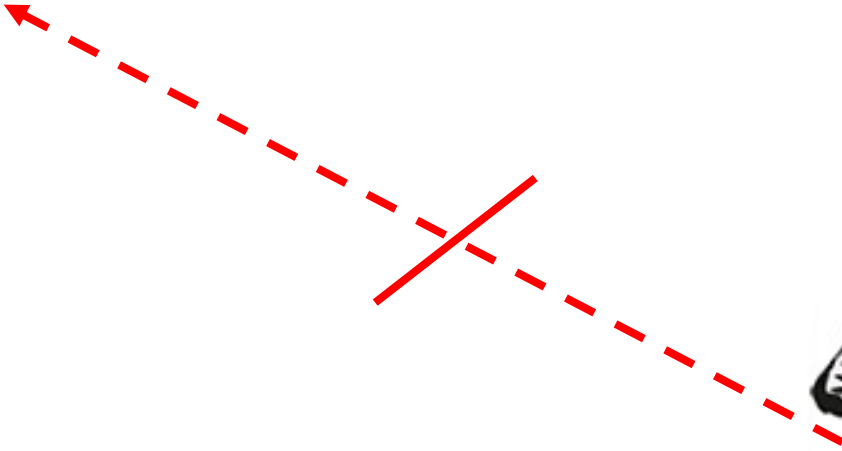
Disentangling the Latent Space

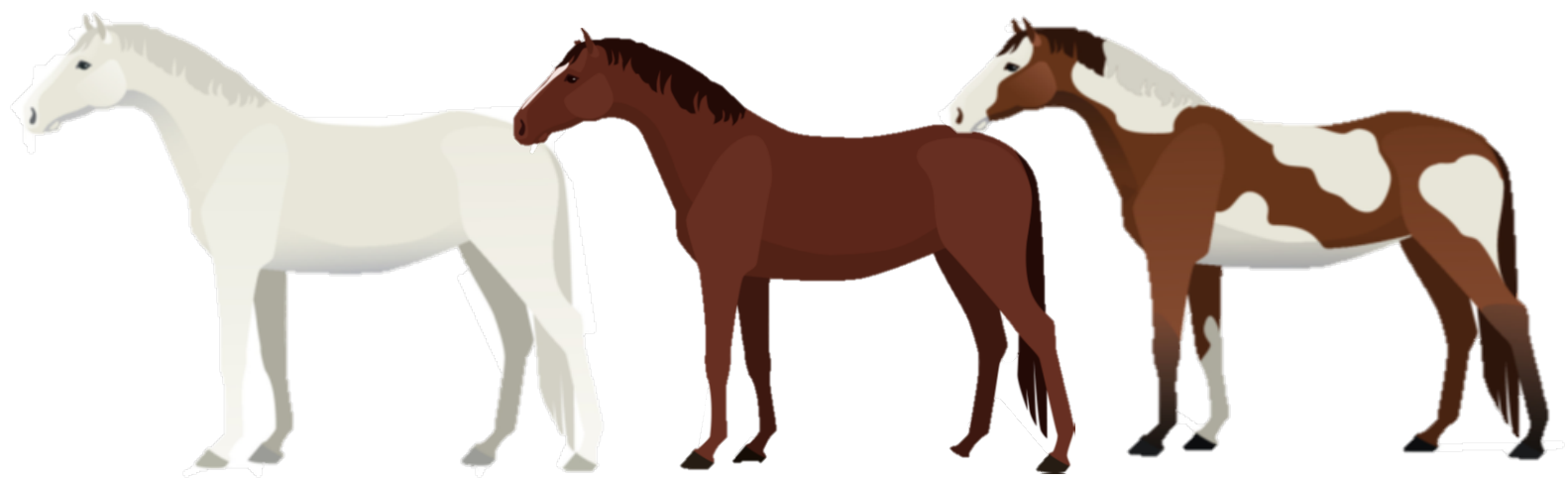
- Multimodal UNIT (MUNIT)
 - A **content** space \mathcal{C} that is **shared, domain-invariant**
 - Two **style** spaces $\mathcal{S}_1, \mathcal{S}_2$ that are **unshared, domain-specific**



Style and Content

- Style: variations within the same domain
(different colors, textures, etc.)
- Content: features shared across two domains

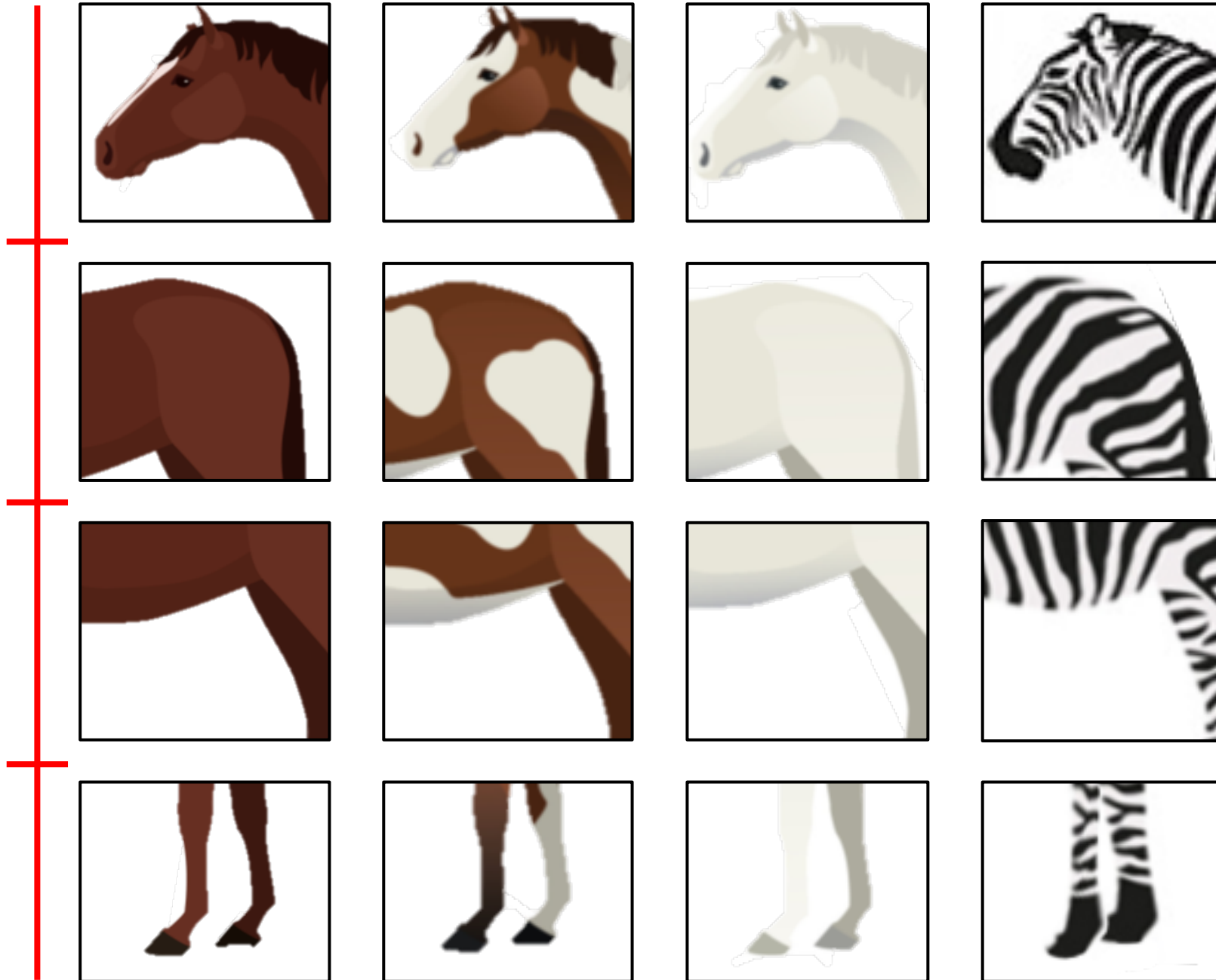




Style

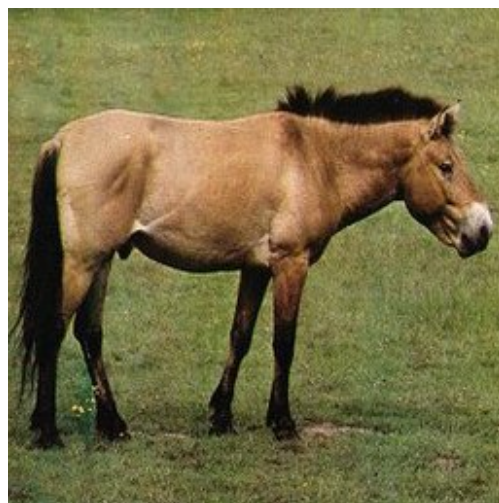


Content

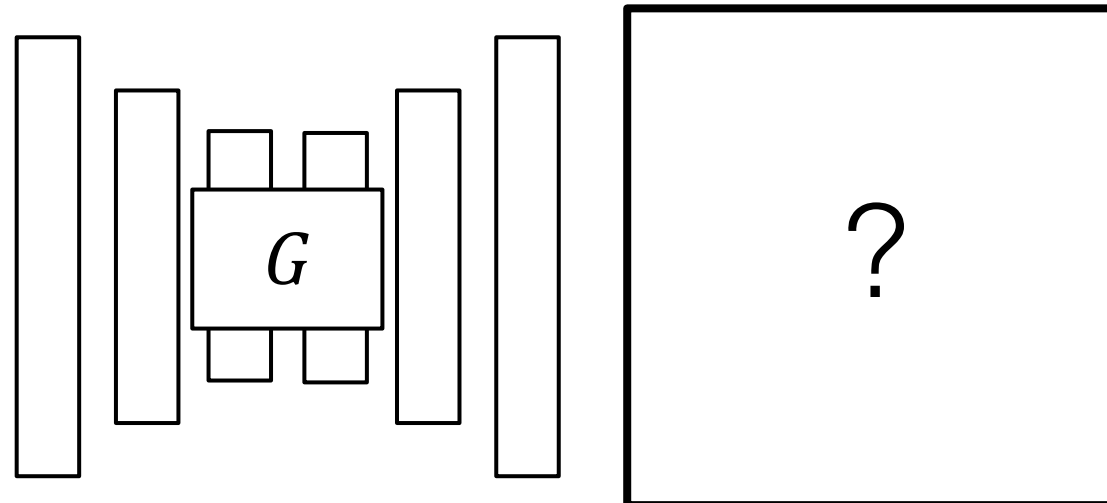


What makes for a good output?

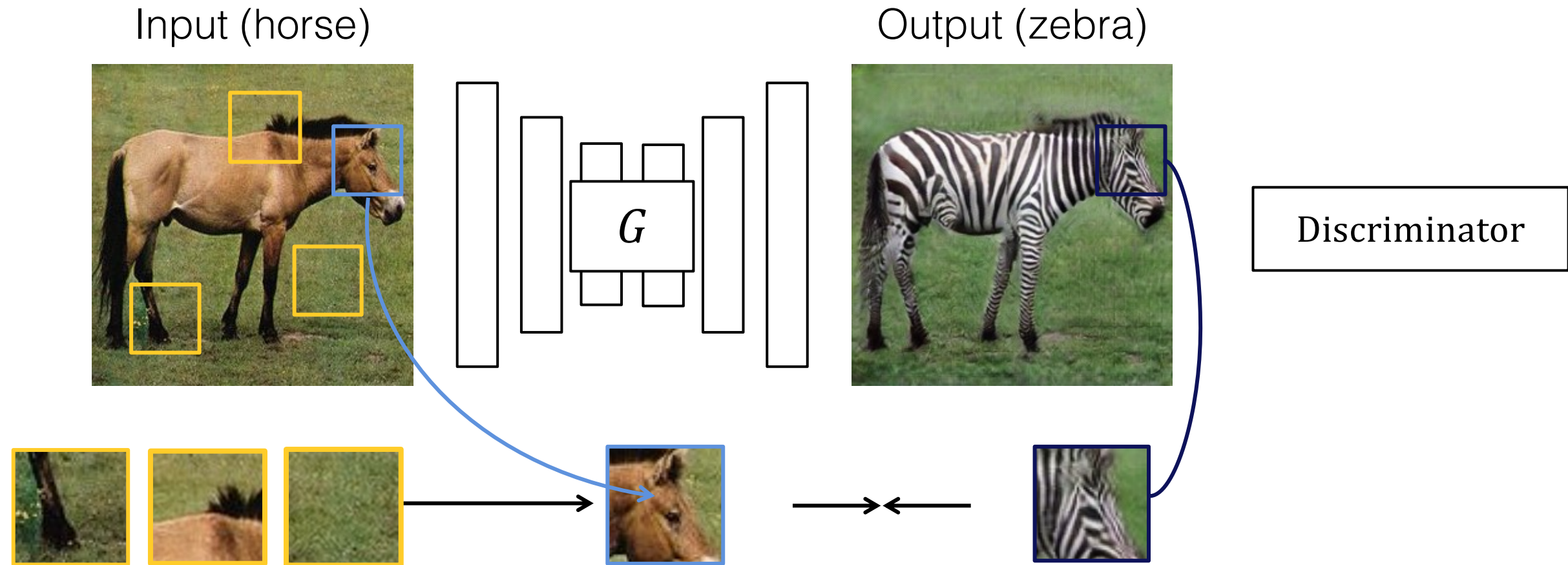
Input (horse)



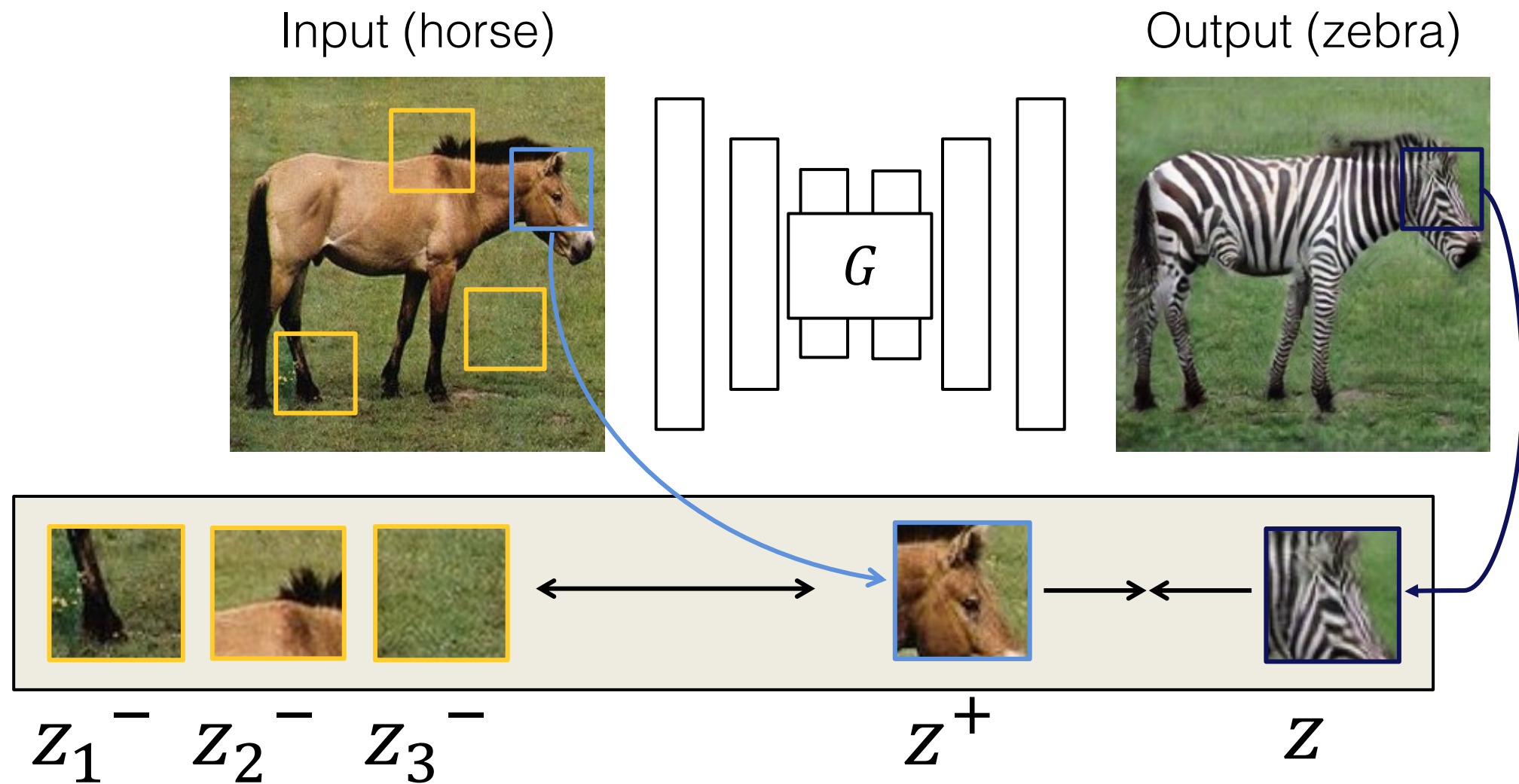
Output (zebra)



Retaining input content

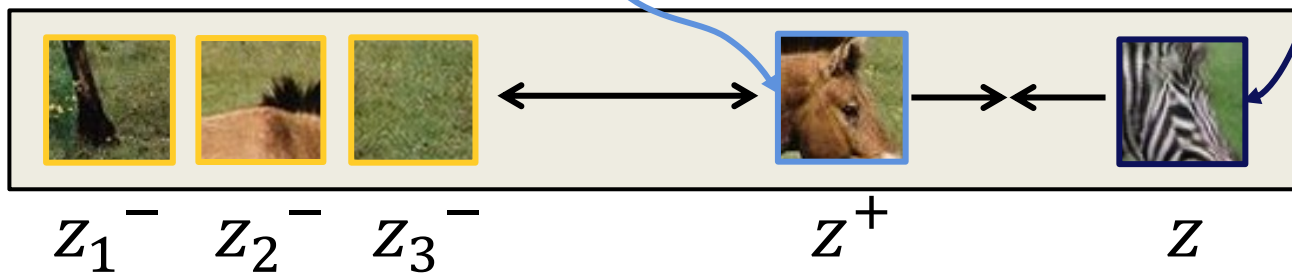
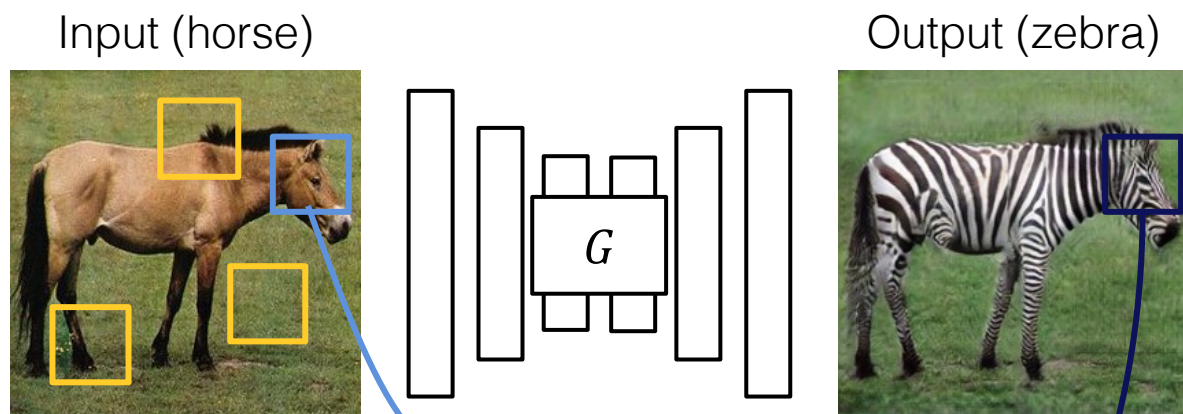


Retaining input content



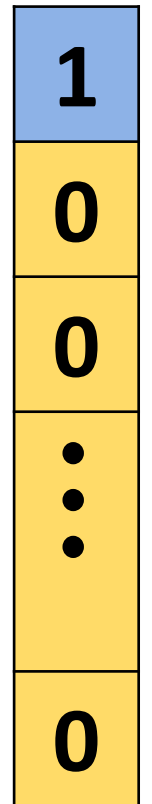
Corresponding patches should have high similarity

Patch-based Contrastive Loss



softmax

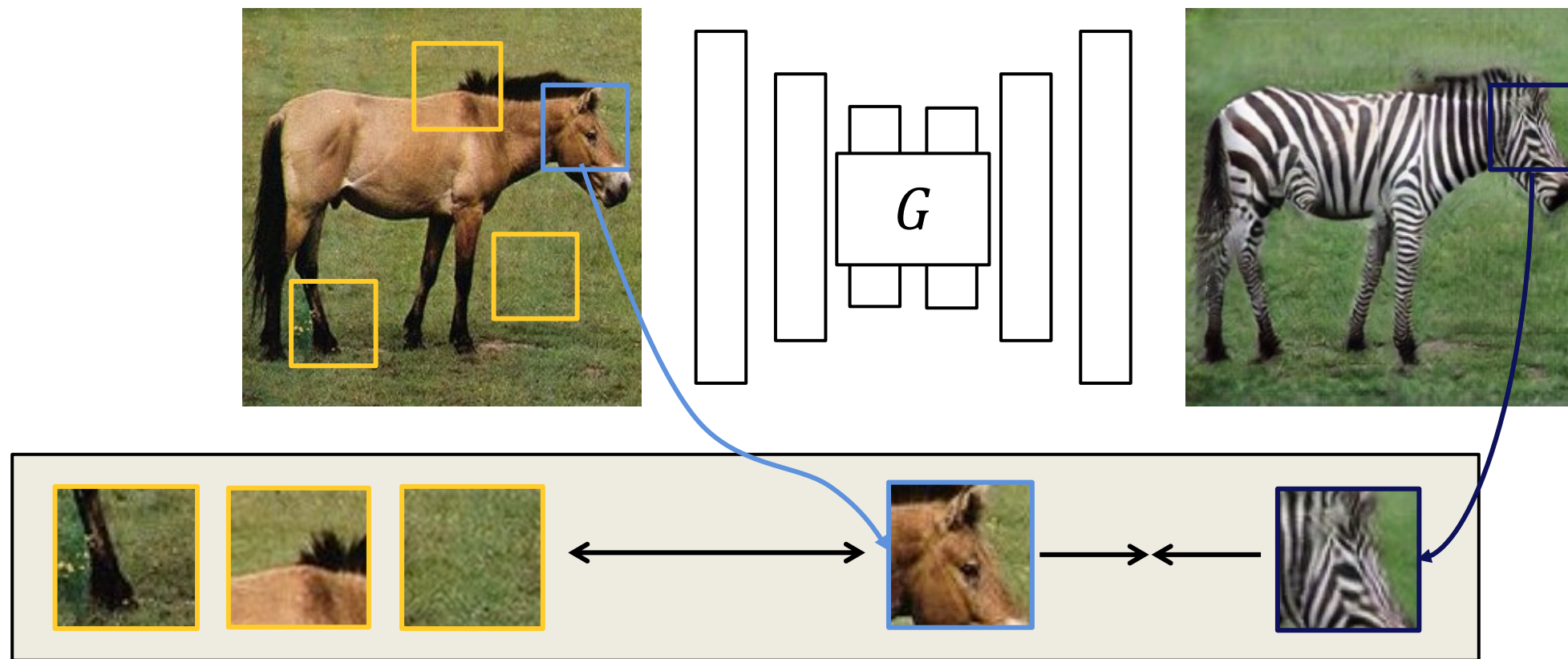
$$\begin{pmatrix} \uparrow z \cdot z^+ / \tau \\ \downarrow z \cdot z_1^- / \tau \\ \downarrow z \cdot z_2^- / \tau \\ \vdots \\ \downarrow z \cdot z_N^- / \tau \end{pmatrix}$$



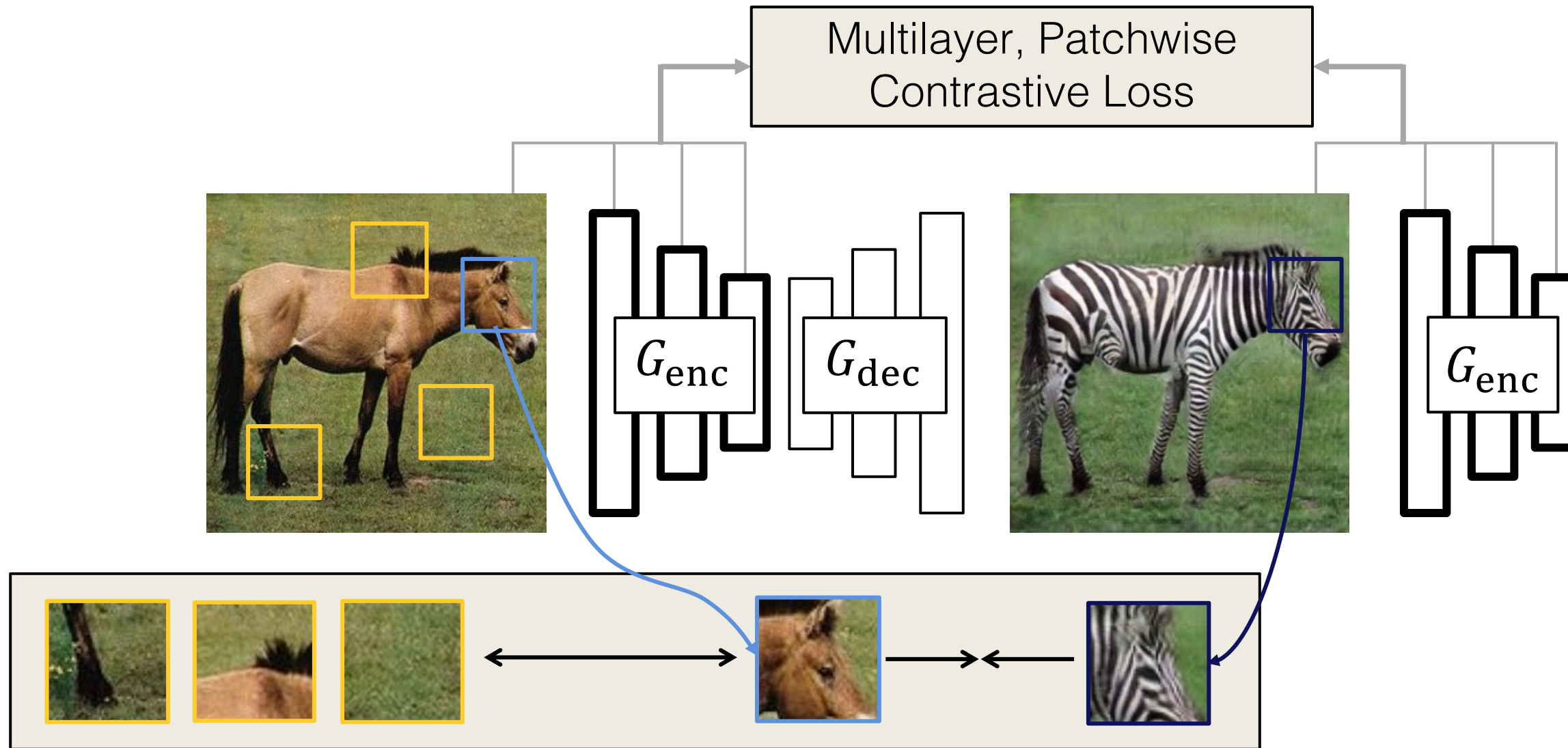
softmax (cosine similarities $/\tau$)
 $\tau=0.07$

- InfoNCE loss (Gutmann et al., AISTATS18 , van den Oord et al., 2018) used in MoCo and SimCLR
- To produce positive pairs:
 - Handcrafted data augmentation (MoCo, SimCLR, etc.)
 - Input and synthesized image (ours)

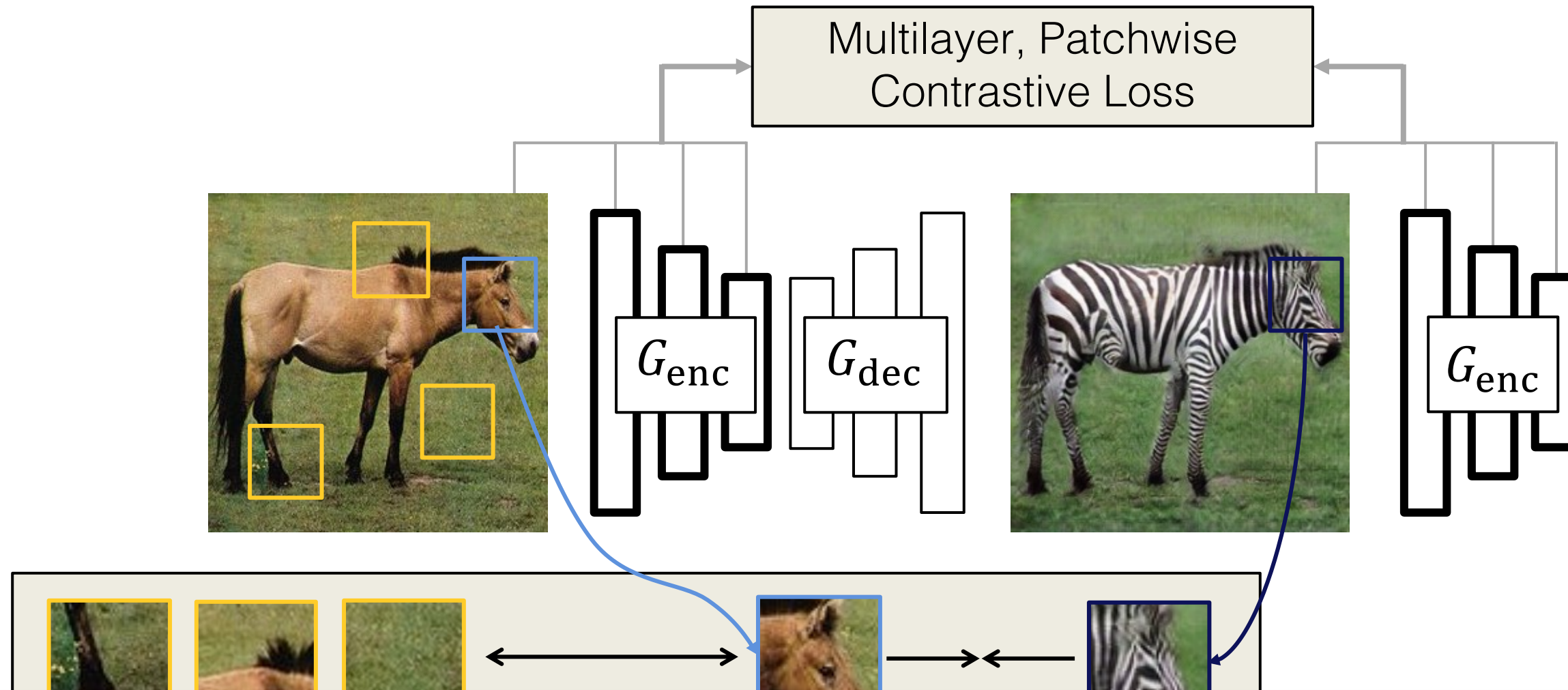
Patchwise contrastive loss



Patchwise contrastive loss

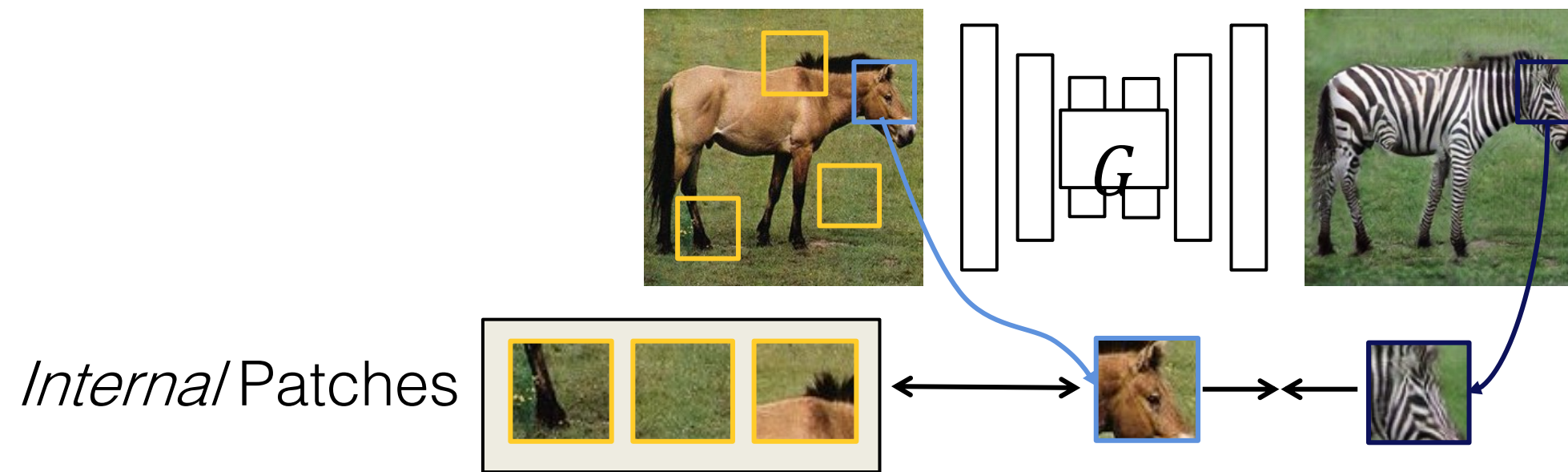


Patchwise contrastive loss

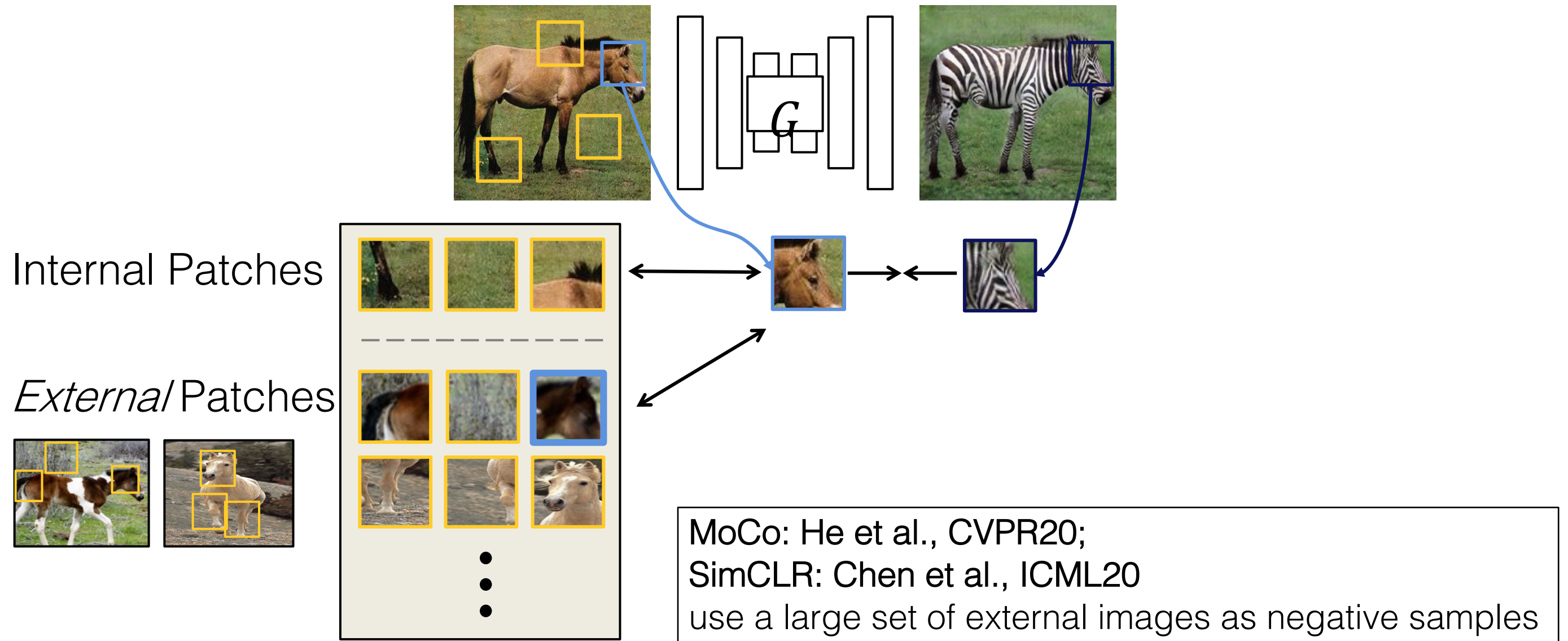


- + No fixed similarity metric (e.g., L1 or perceptual loss)
- + One-sided (no inverse mapping needed)

Internal vs External Patches



Internal vs External Patches



External patches make things worse

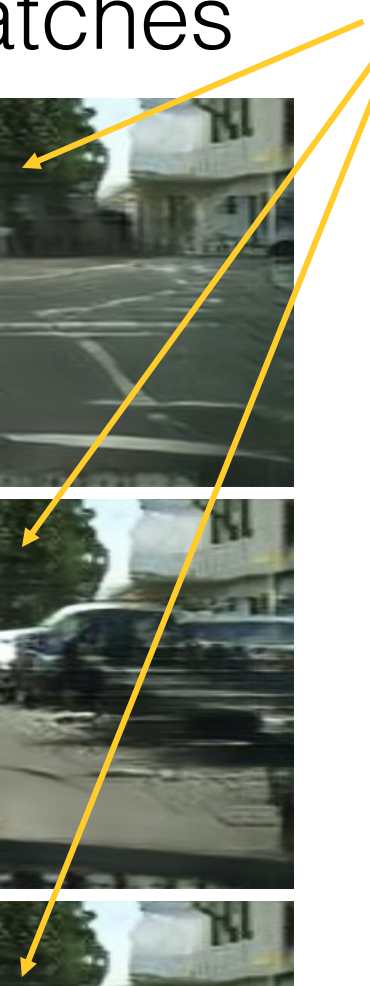
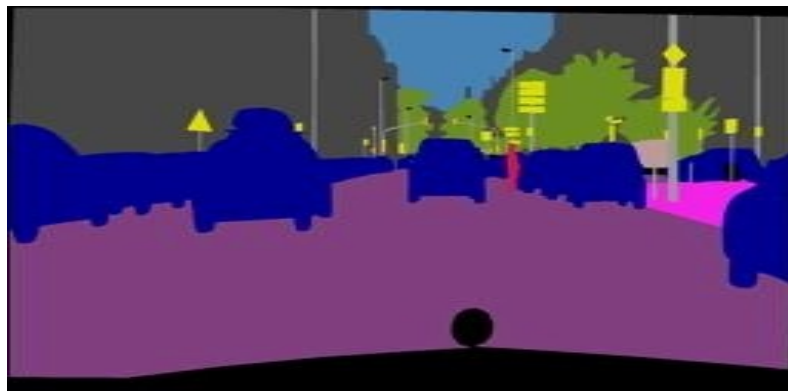
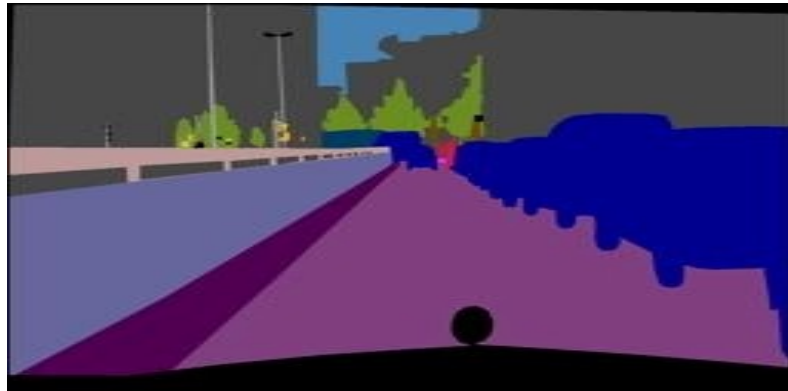
Internal vs External Patches

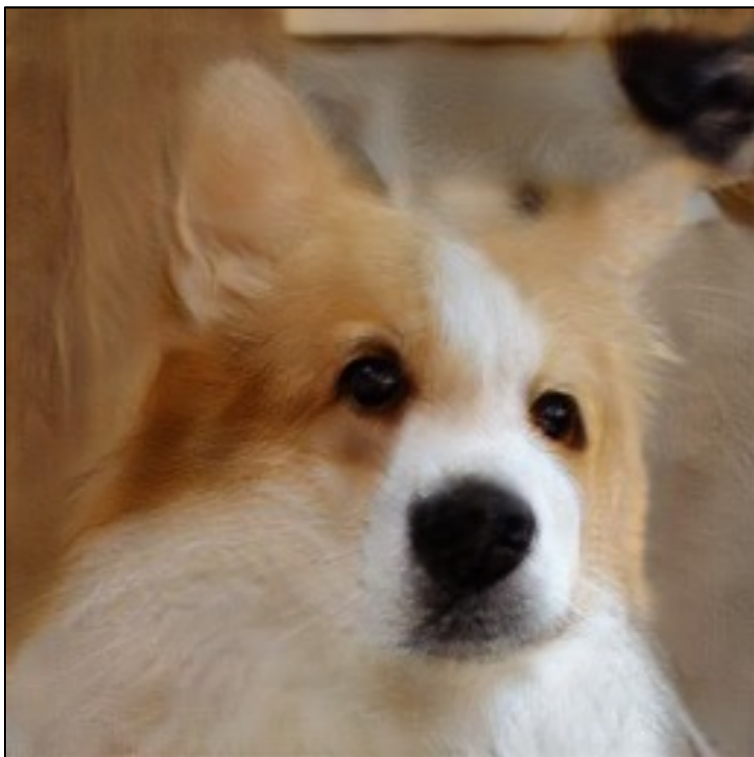
input

internal patches

external patches

**Mode
Collapse!**





Cat



Yosemite Summer



Apple



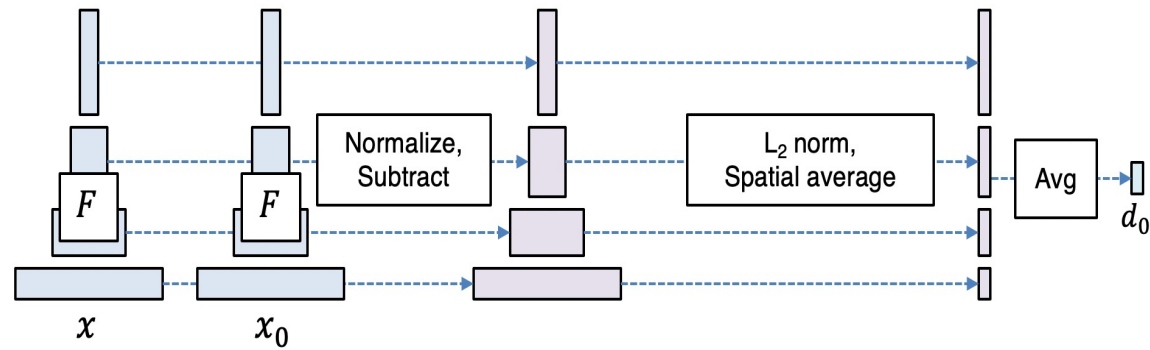
Paris



GTA

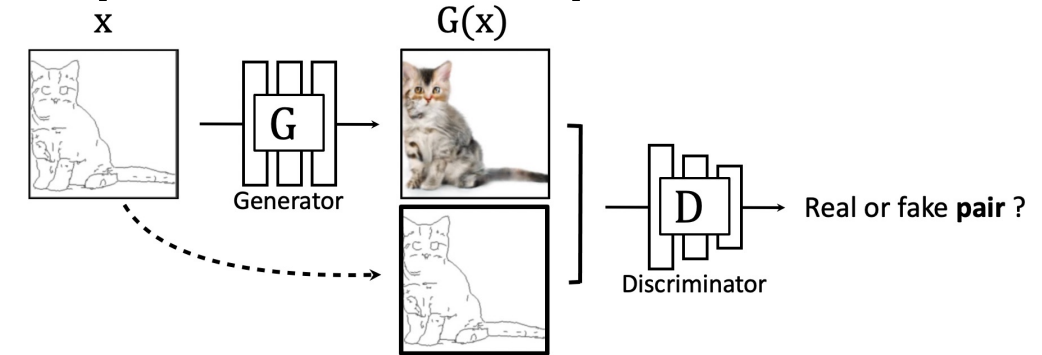
Review

Perceptual/Feature Loss



How well do "perceptual losses" describe perception?

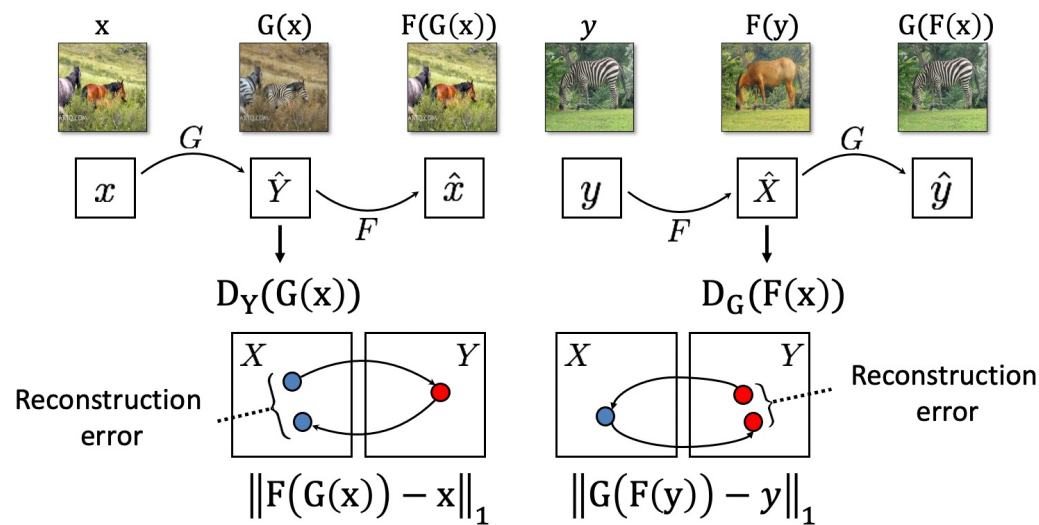
(Conditional) GAN Loss



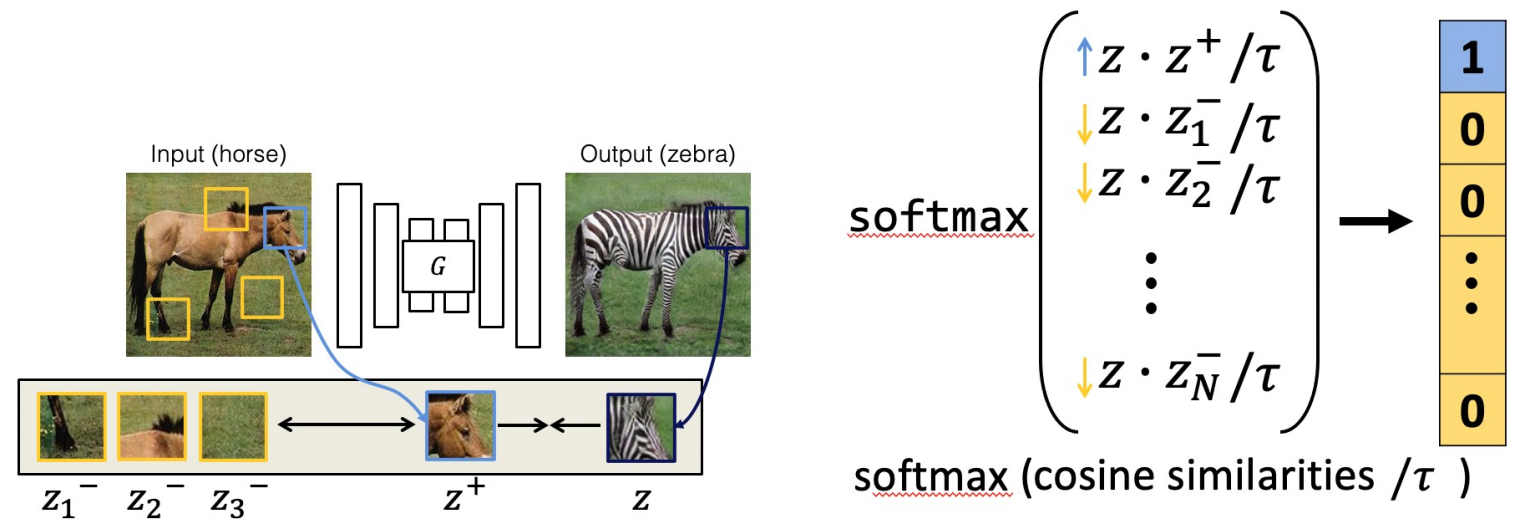
Learning objective

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

Cycle-Consistency Loss



Patch-wise Contrastive Loss



Other loss functions: Style Loss [Gatys et al.], Contextual Loss [Mechrez et al.], Domain-specific Loss (e.g., face), 3D-aware Loss (for geometric data)



Style and Content, Texture Synthesis

Jun-Yan Zhu

16-726, Spring 2022

Texture

- Texture depicts spatially repeating patterns
- Many natural phenomena are textures



radishes



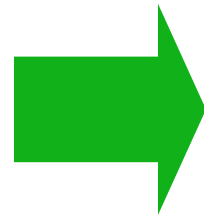
rocks



yogurt

Texture Synthesis

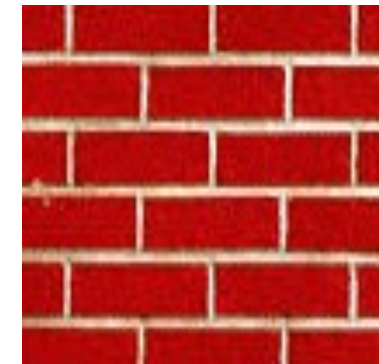
- Goal: create new samples of a given texture
- Applications: virtual environments, inpainting, texturing surfaces



Non-parametric Texture Synthesis

The Challenge

- Need to model the whole spectrum: from repeated to stochastic texture



repeated

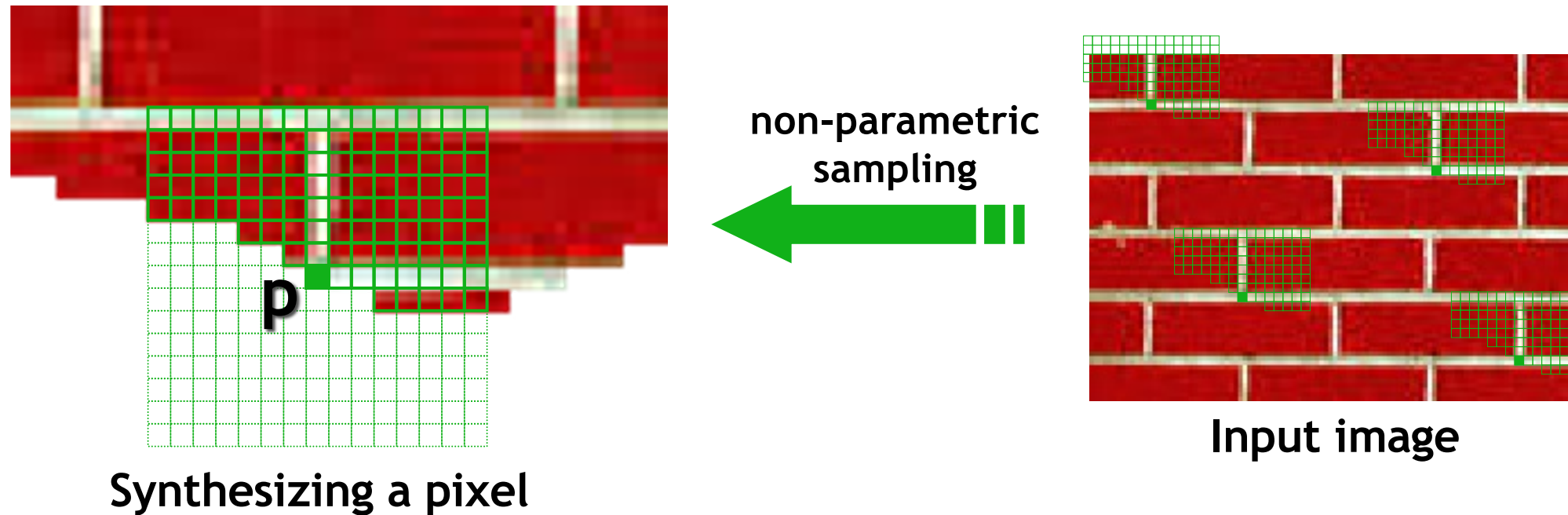


stochastic



Both?

Efros & Leung Algorithm



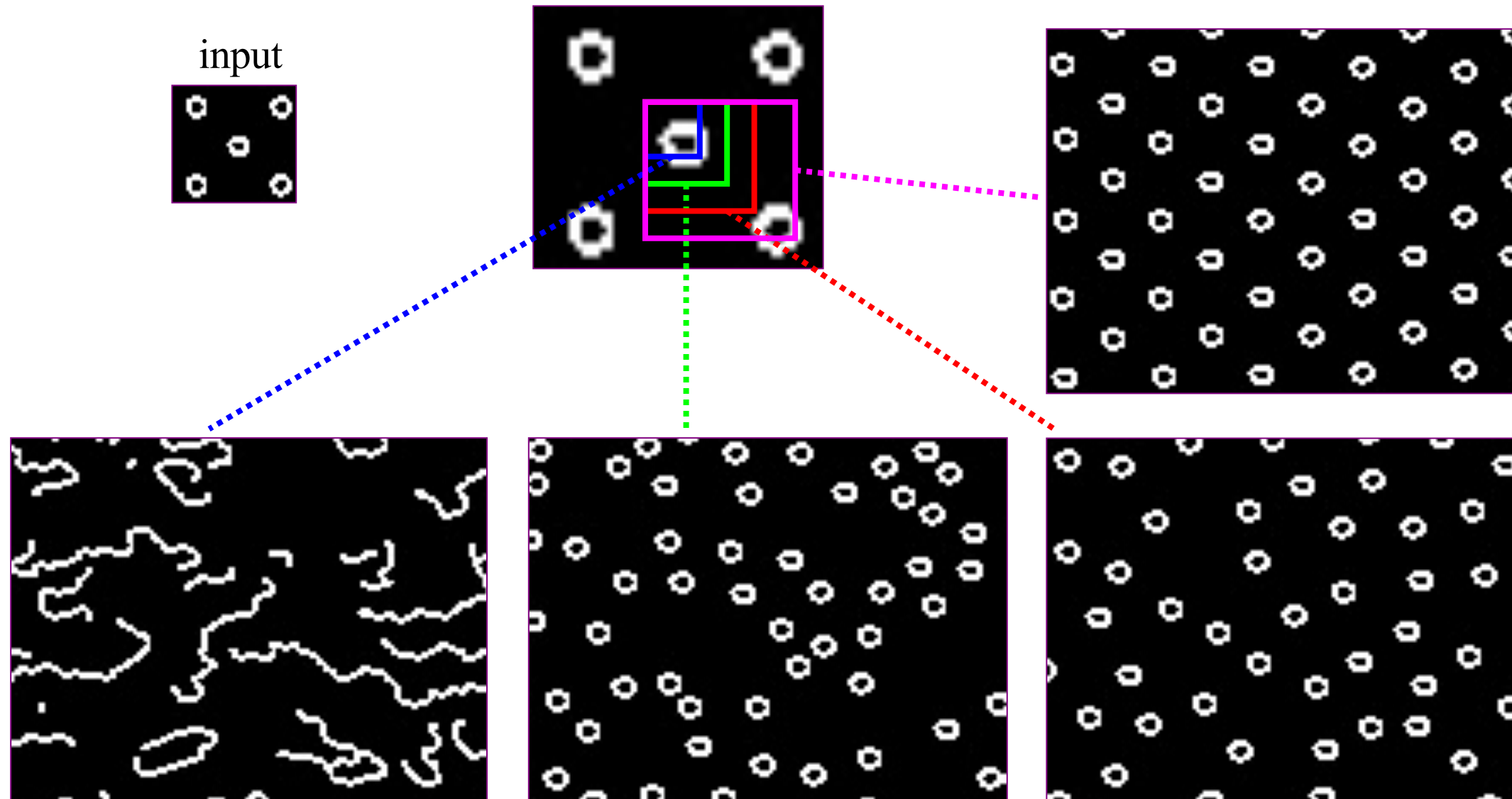
- Assuming Markov property, compute $P(\mathbf{p} | N(\mathbf{p}))$
 - Building explicit probability tables infeasible
 - Instead, we *search the input image* for all similar neighbourhoods — that's our pdf for \mathbf{p}
 - To sample from this pdf, just pick one match at random

PixelCNN

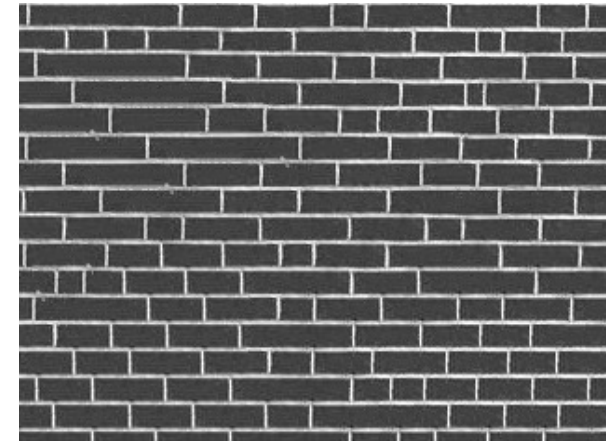
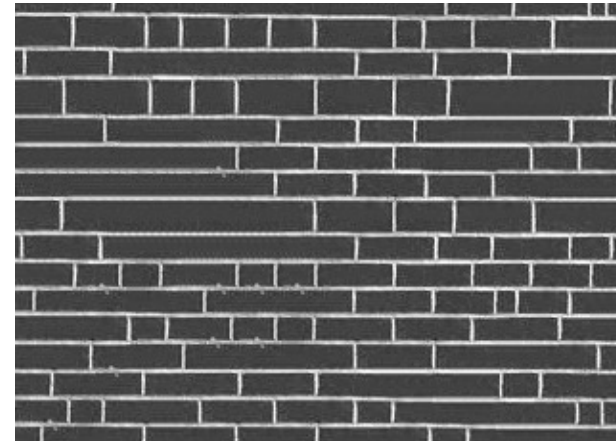
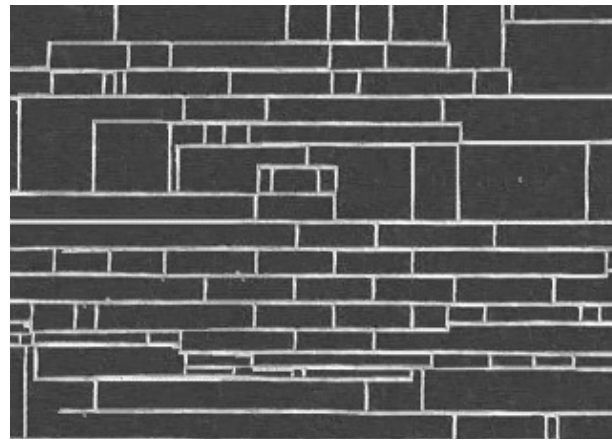
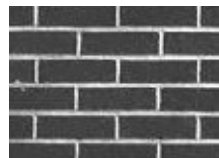
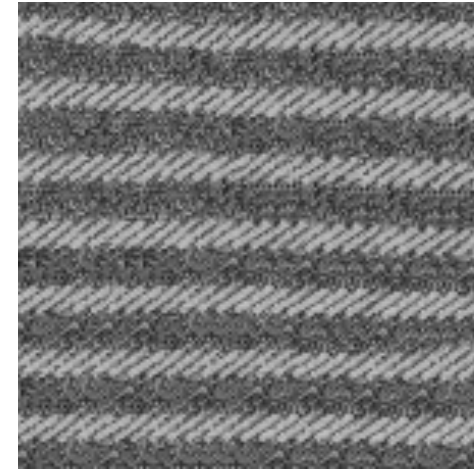
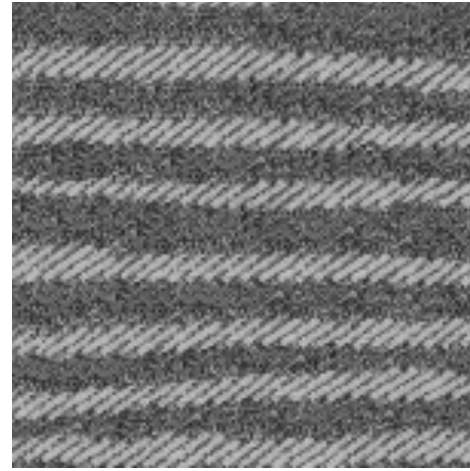
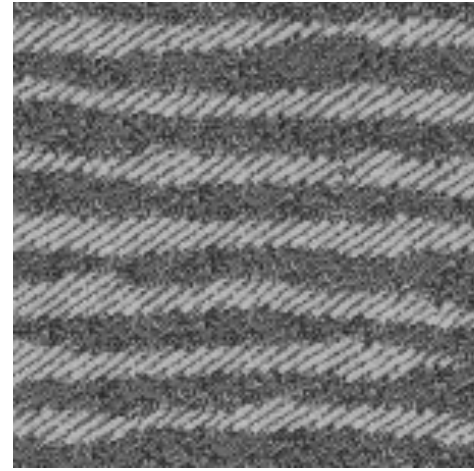
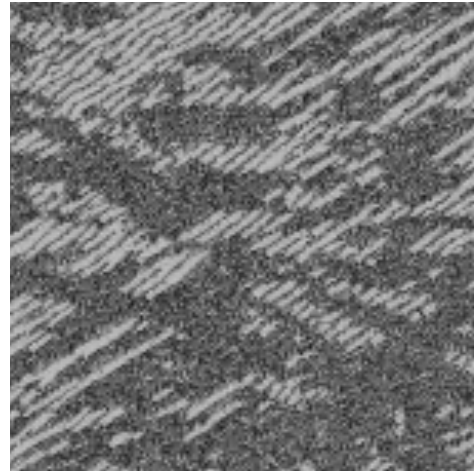
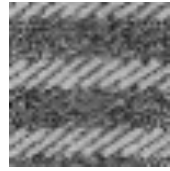
Some Details

- Growing is in “onion skin” order
 - Within each “layer”, pixels with most neighbors are synthesized first
 - If no close match can be found, the pixel is not synthesized until the end
- Using *Gaussian-weighted SSD* is very important
 - to make sure the new pixel agrees with its closest neighbors
 - Approximates reduction to a smaller neighborhood window if data is too sparse

Neighborhood Window



Varying Window Size

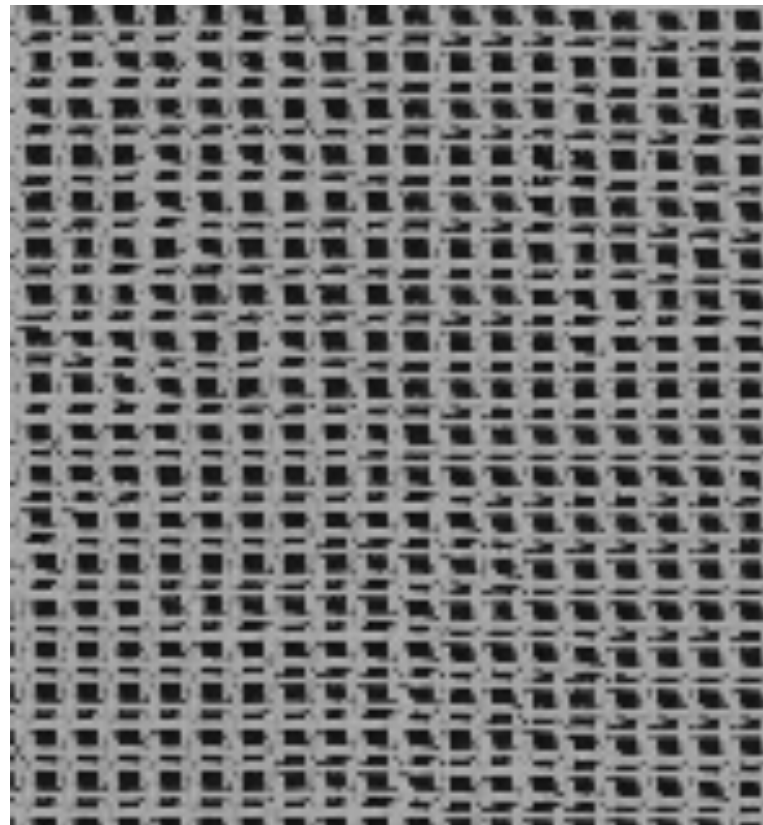
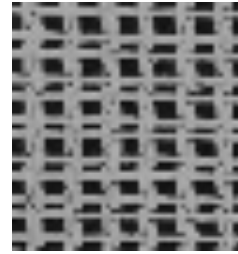


Increasing window size

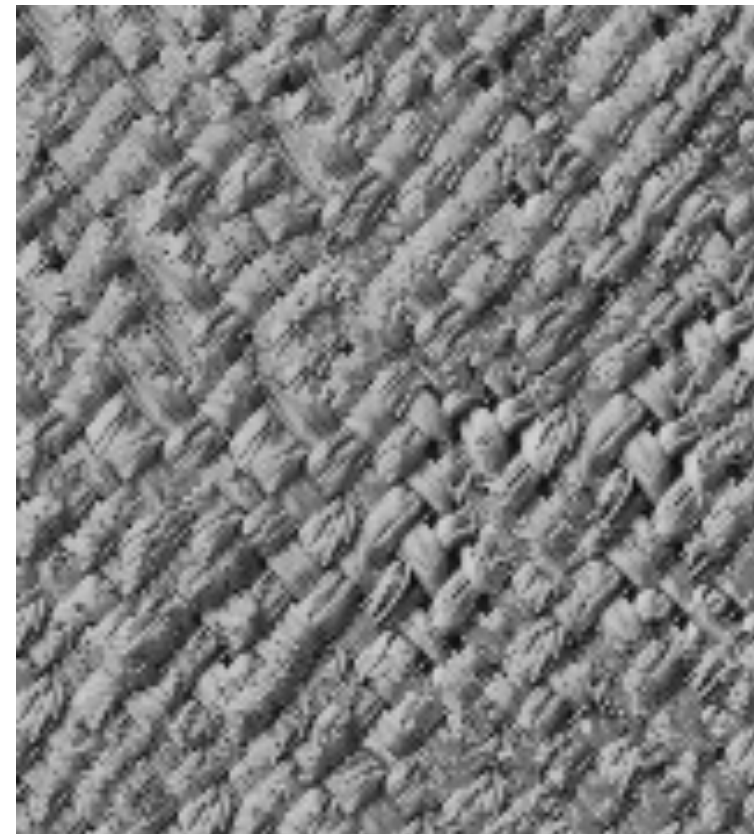
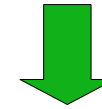


Synthesis Results

french canvas



rafia weave

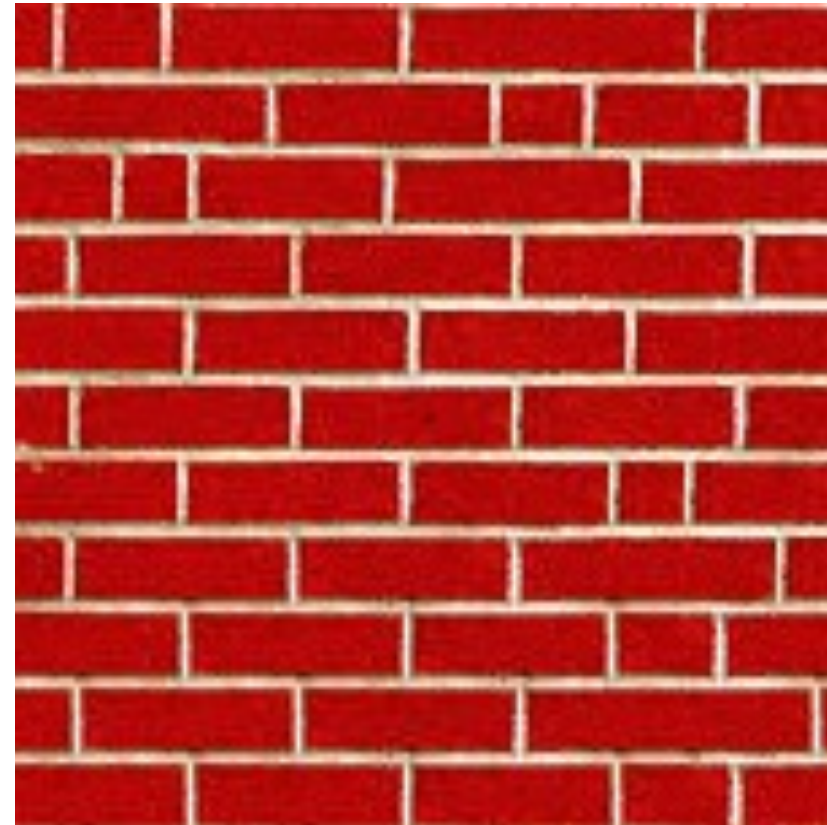
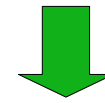


More Results

white bread

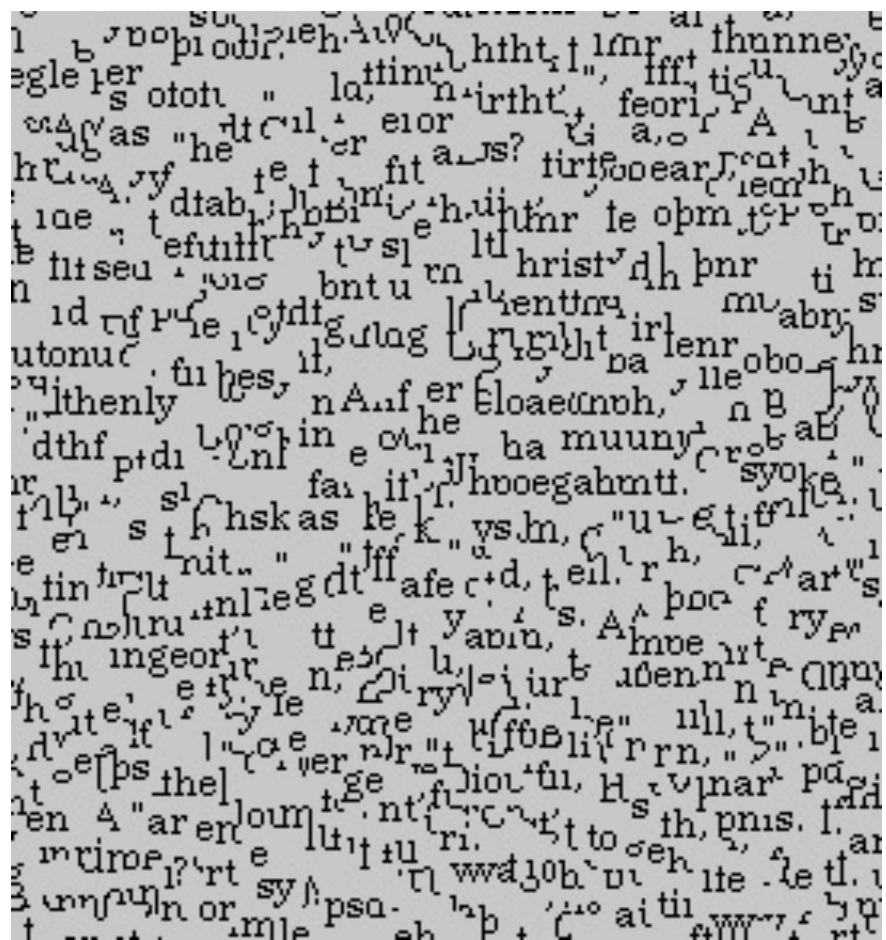


brick wall



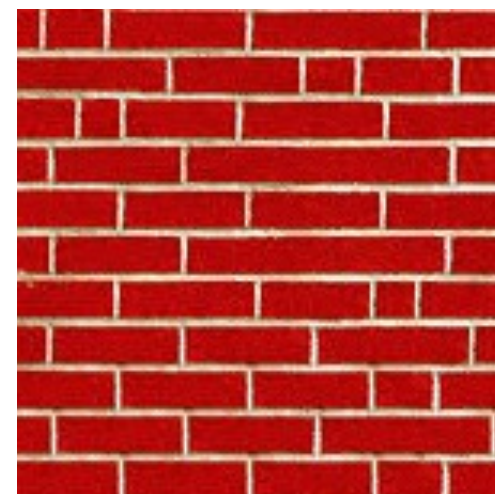
Homage to Shannon

...coming in the unsensational
...r Dick Gephardt was fair
...rful riff on the looming
...nly asked, "What's your
...tions?" A heartfelt sigh
...story about the emergen
...es against Clinton. "Boy
...g people about continuin
...ardt began, patiently obs
...s, that the legal system h
...g with this latest tanger

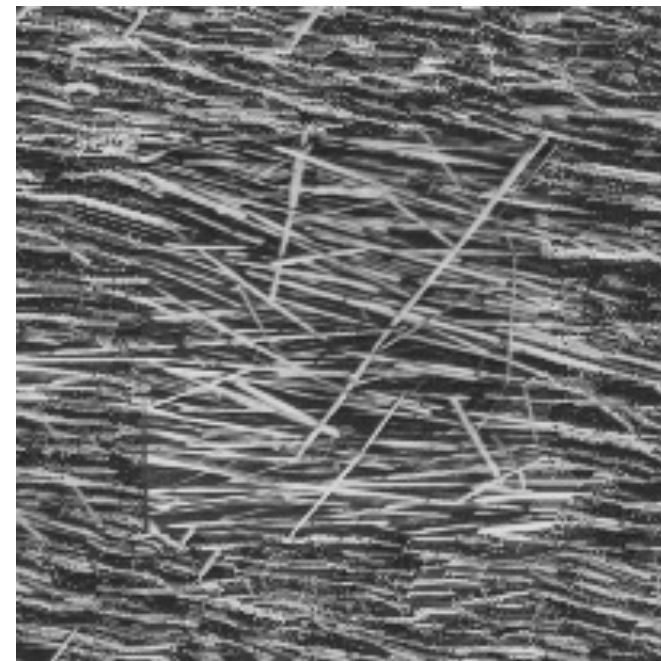
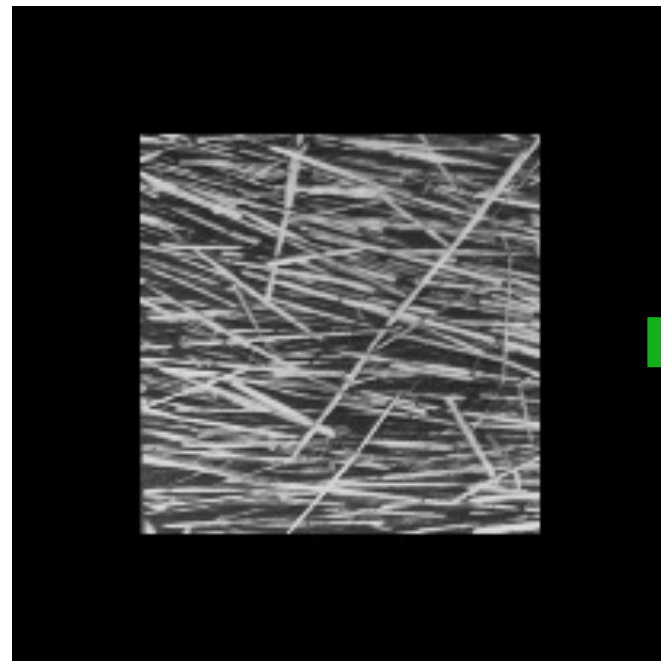


...athaim. them . "Whephartfe bartifelintomimen
...el ck Clirtioout omaim thartfelins.f out s ane
...the ry onst wartfe lck Gephtoomimeationl sigab
...Chiooufit Clinut Cil riff on. hat's yo'dn, parut tly
...ons yontonsteht waked, paim t sahe loo riff on
...nskoneploourtfeas leil A nst Clit, "Wleontongal s
...k Cirtioouirtfepe ong pme abegal fartfenstemem
...tiensteneltorydt telemephinsverdt was agemer
...ff ons artientont Cling peme as urtfe atich, "Boui s
...nal s fartfelt sig pedrt h'dt ske abounutie aboutioo
...tfaonewas you aboronthardt thatins fain, ped, '
...ains. them, pabout wasy arfint coutly d, l n A h
...ble emthringbooreme agas fa bontinsyst Clinut
...ory about continst Clipseopinst Cloke agatiff out
...stome zinemen tly ardt beoraboul n, thenly as t
...cons faimeme Diontont wat coutlyohgans as fan
...ien, phrtfaul, "Wbout cout congagal comininga
...mifmst Clity abon al coountha.emungairt tfoun
...The loocrysta loontieph. intly on, theoplegatick
...ul tatieontly atie Diontiomt wal s f tbegea ener
...nthahgat's enenhmas fan. "intchthorv ahons v

Hole Filling



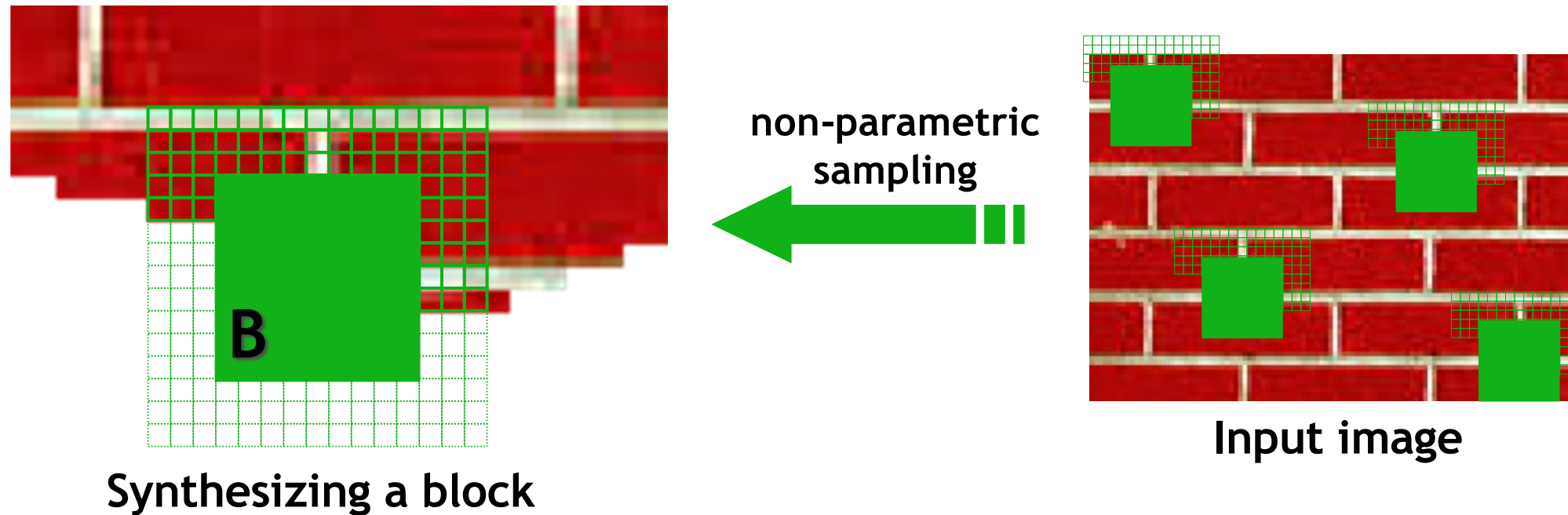
Extrapolation



Summary

- The Efros & Leung algorithm
 - + Very simple
 - + Surprisingly good results
 - + Synthesis is easier than analysis!
 - ...but very slow

Image Quilting [Efros & Freeman]

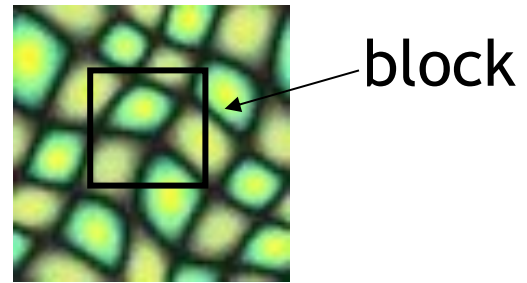


- Observation: neighbor pixels are highly correlated

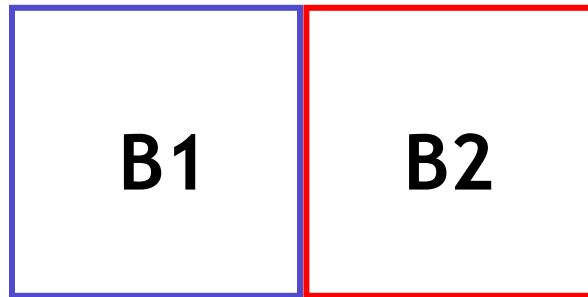
Idea: unit of synthesis = block

- Exactly the same but now we want $P(\mathbf{B} | N(\mathbf{B}))$
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!

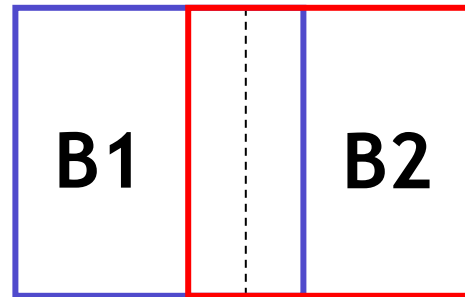
VQ-VAE2



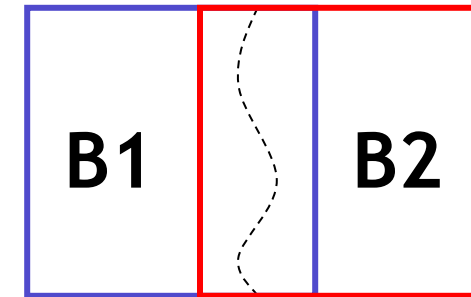
Input texture



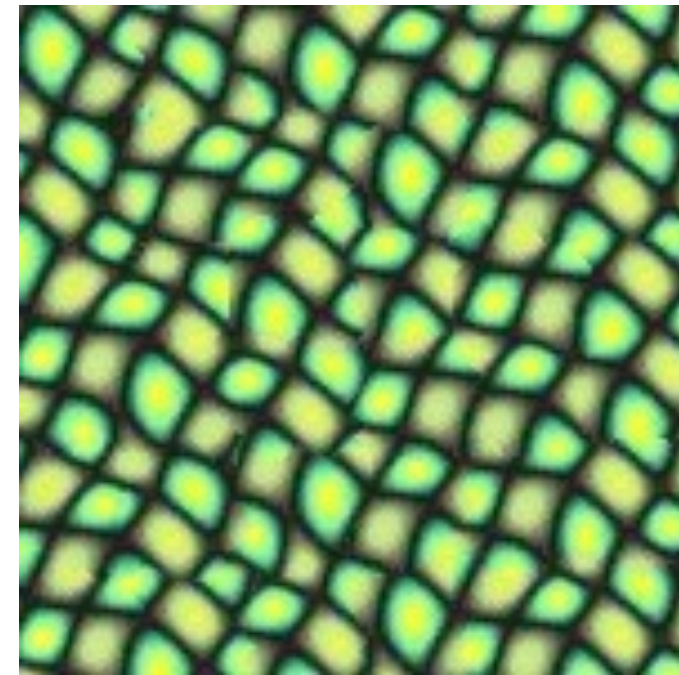
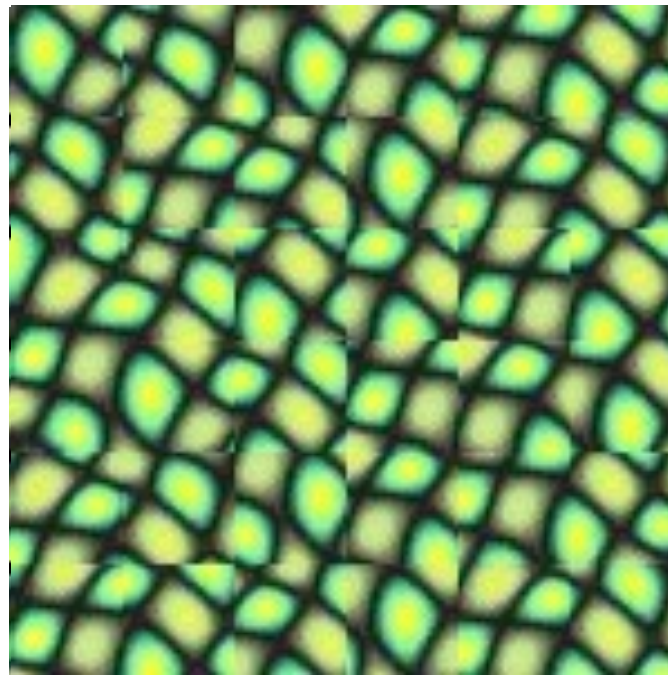
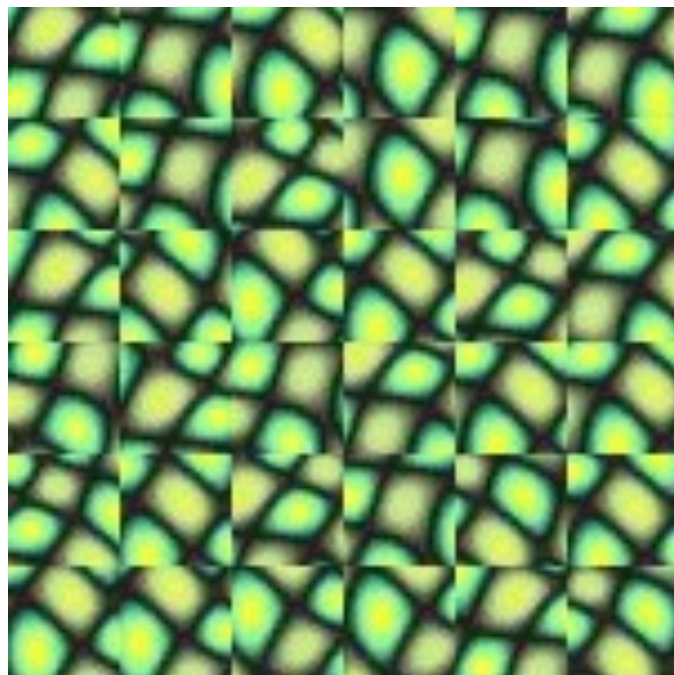
Random placement of blocks



Neighboring blocks constrained by overlap

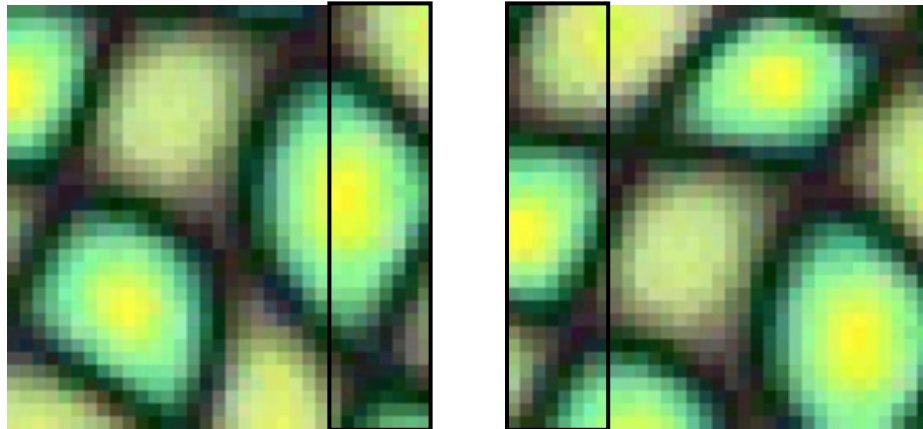


Minimal error boundary cut

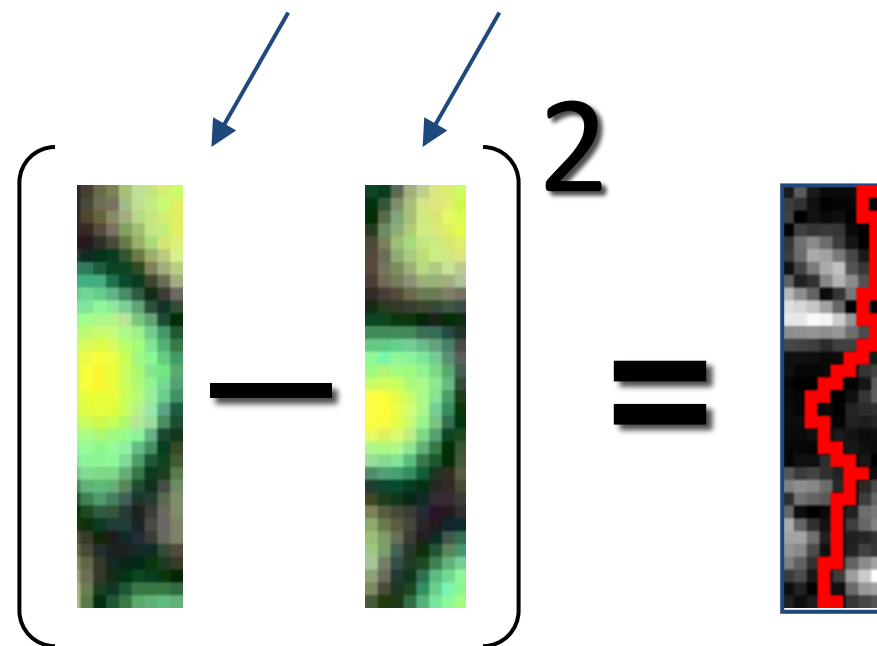
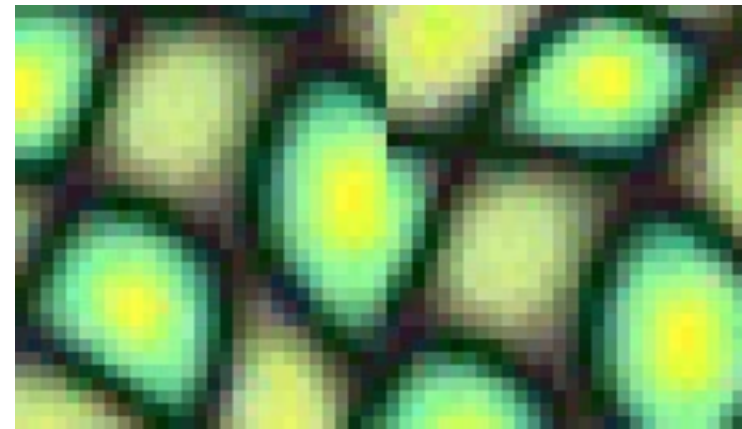


Minimal error boundary

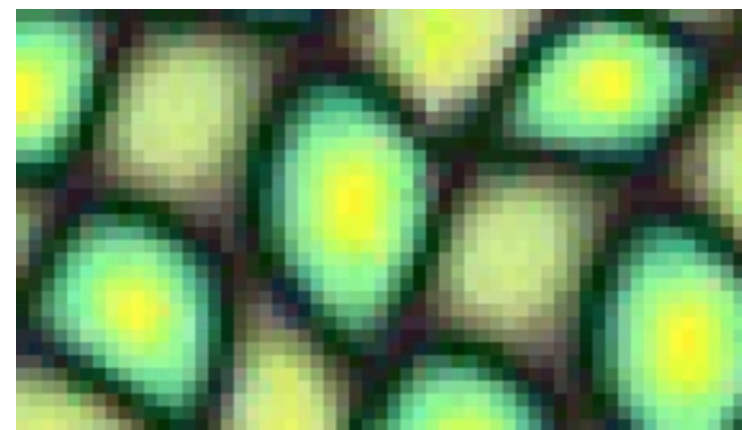
overlapping blocks



vertical boundary



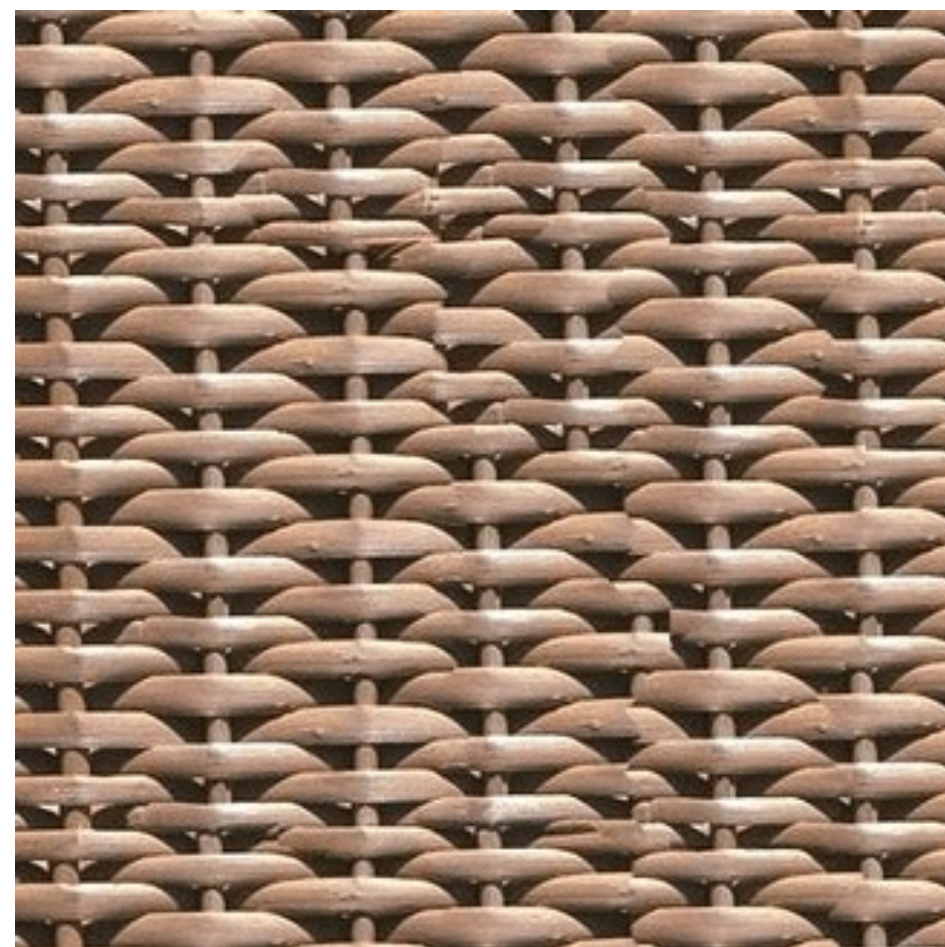
overlap error

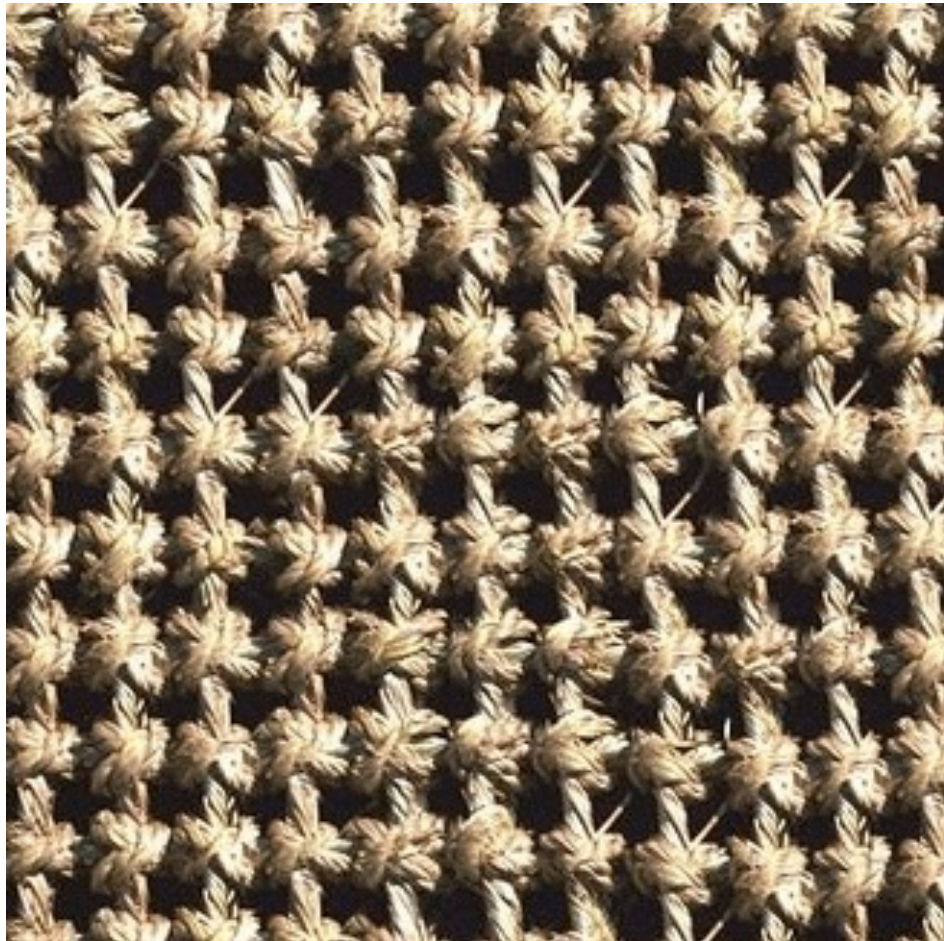


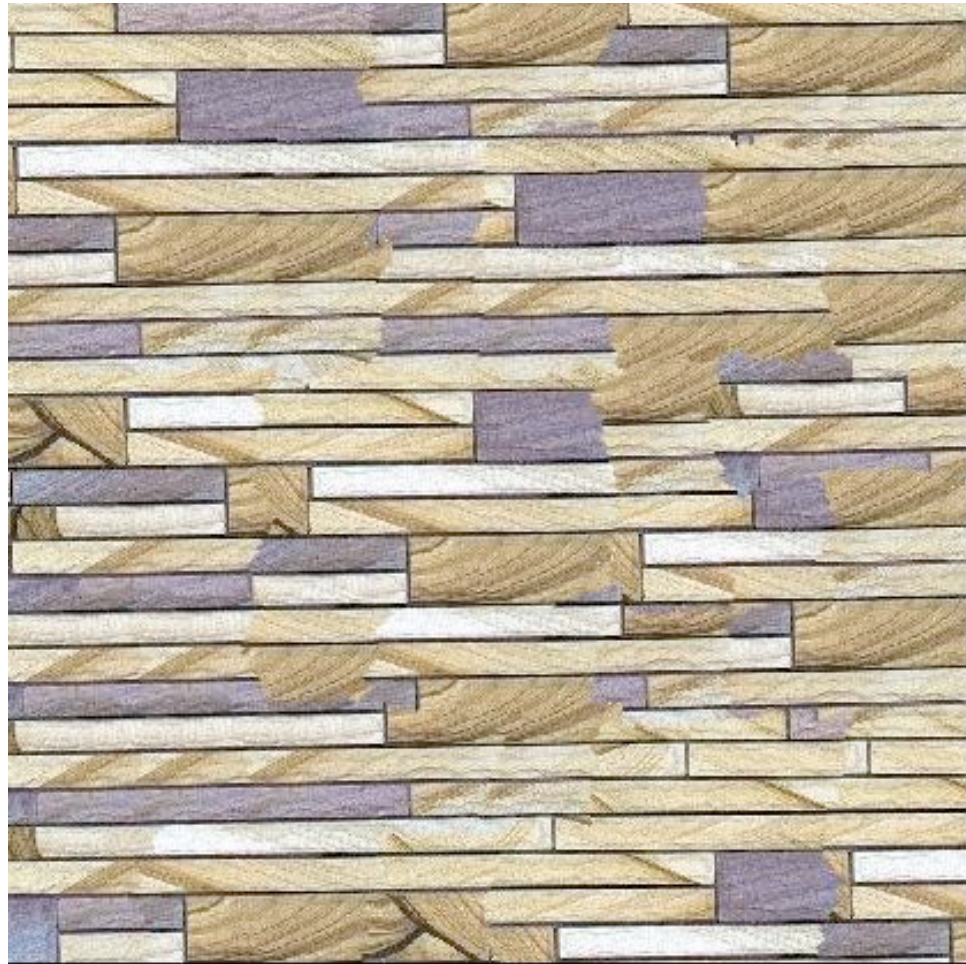
min. error boundary

Our Philosophy

- The “Corrupt Professor’s Algorithm”:
 - Plagiarize as much of the source image as you can
 - Then try to cover up the evidence
- Rationale:
 - Texture blocks are by definition correct samples of texture so problem only connecting them together

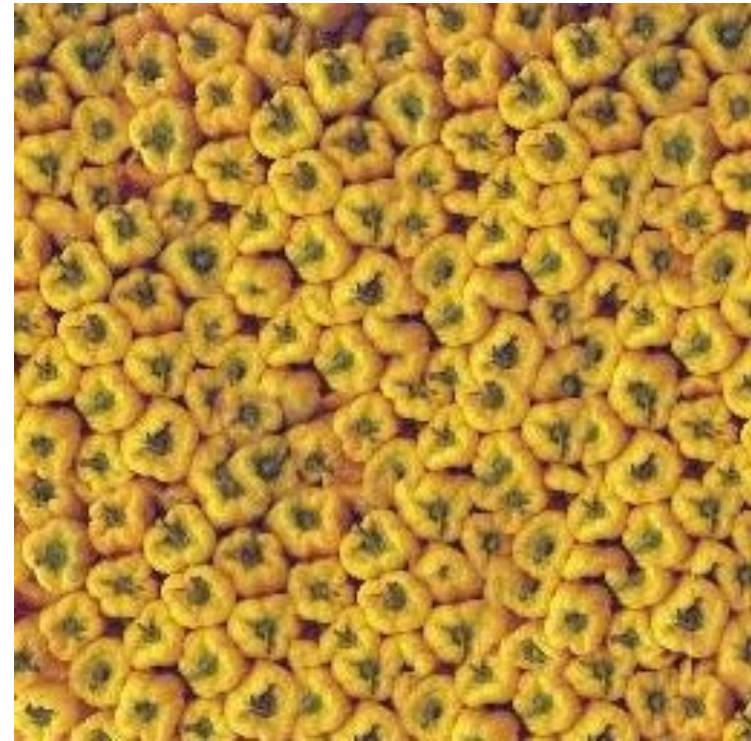
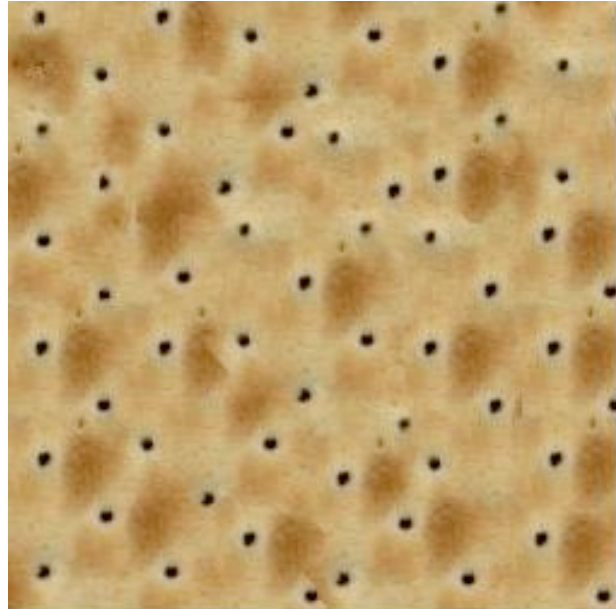
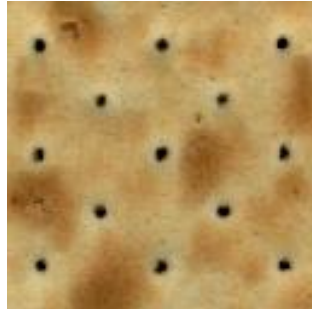






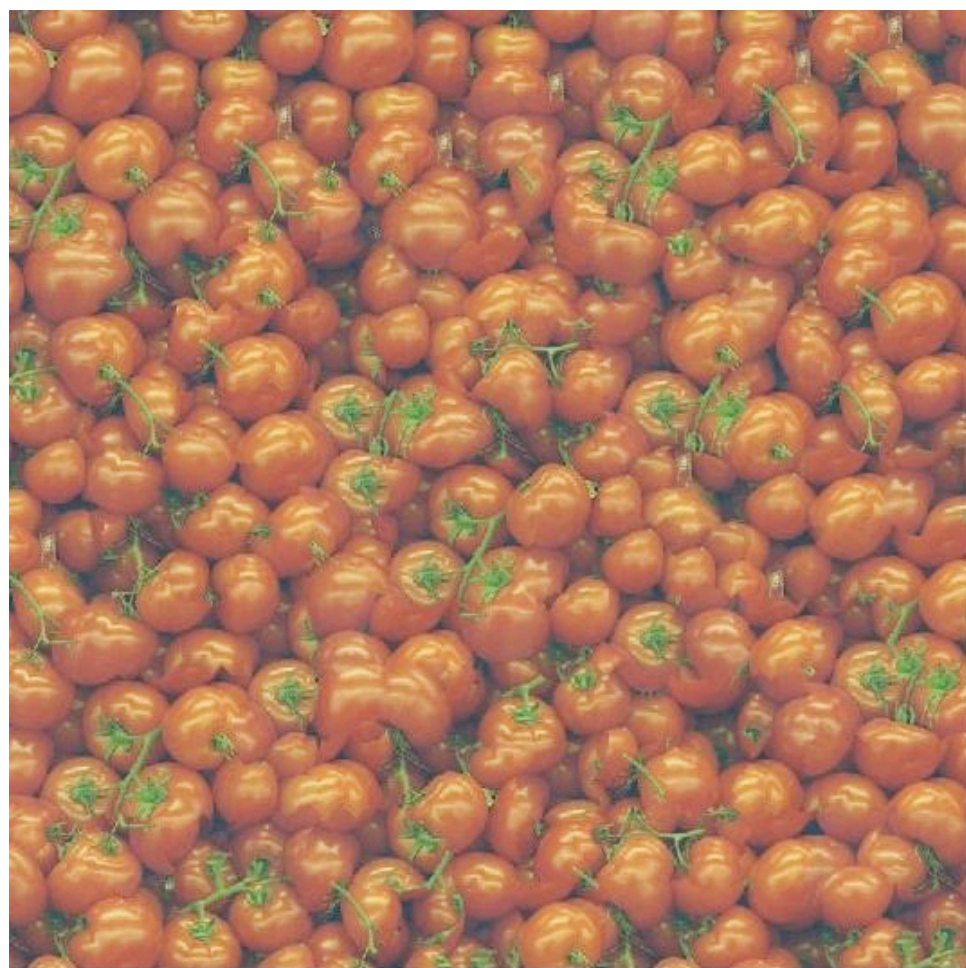


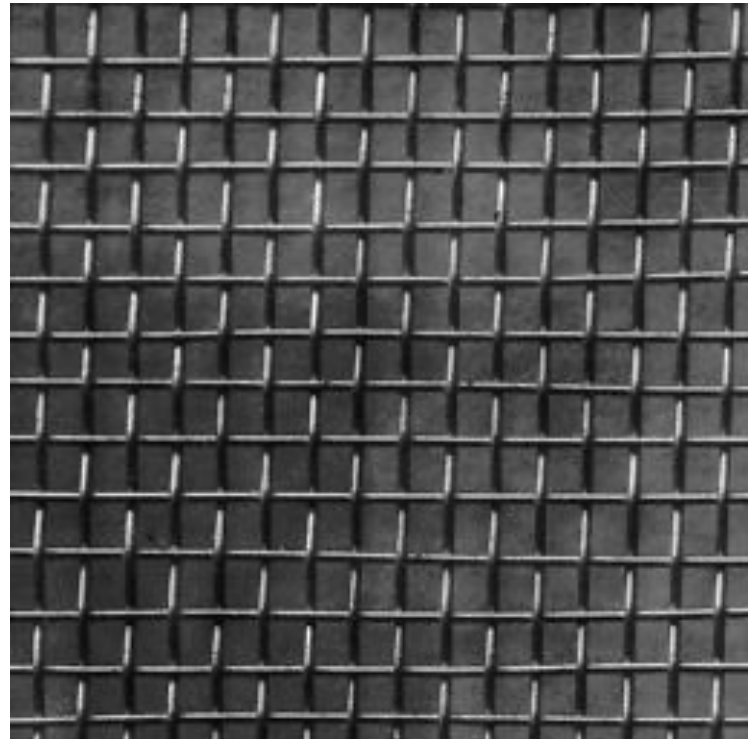




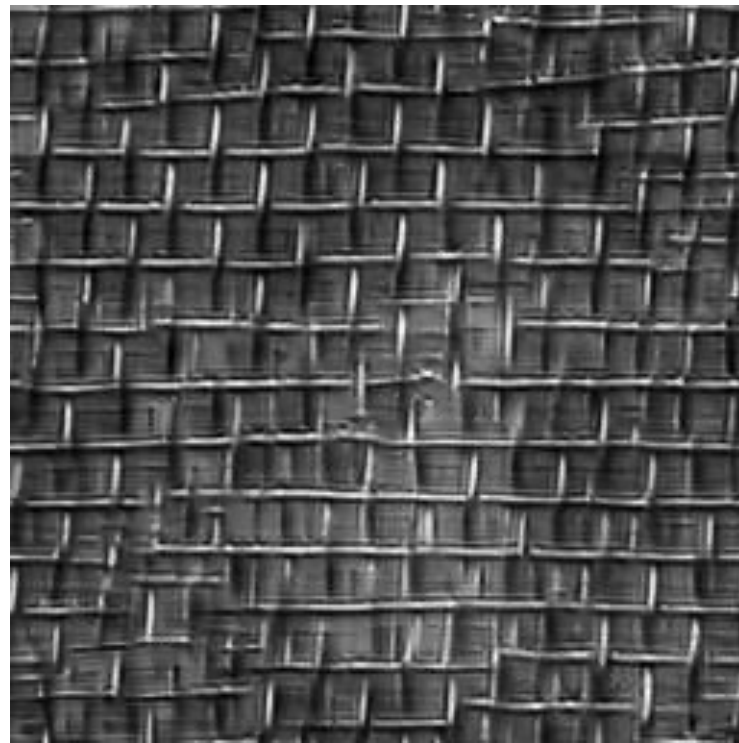


Failures (Chernobyl Harvest)

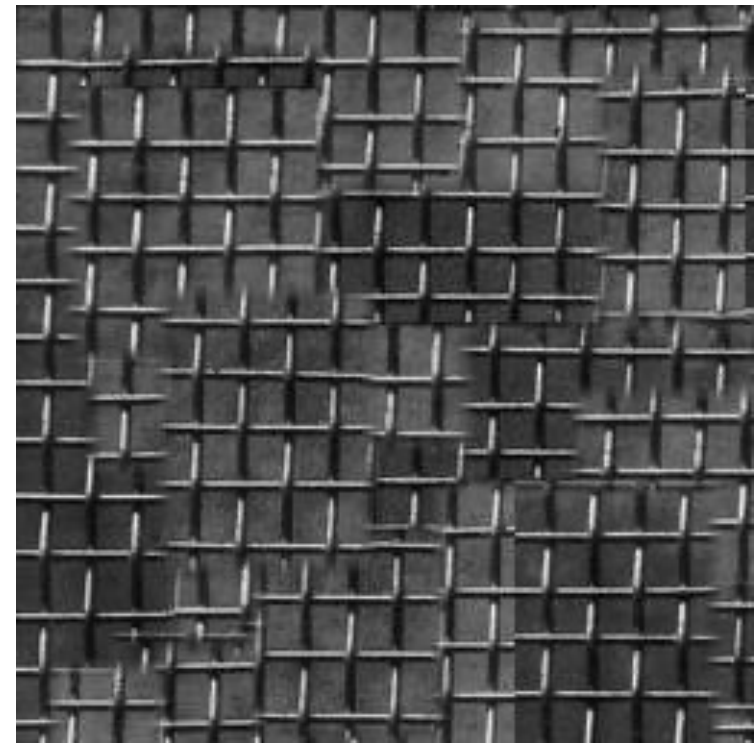




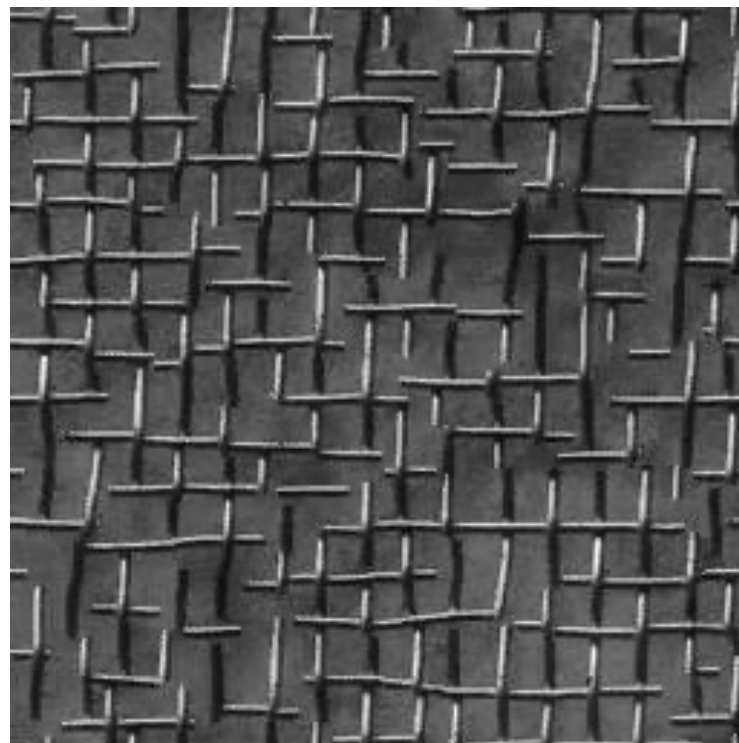
input image



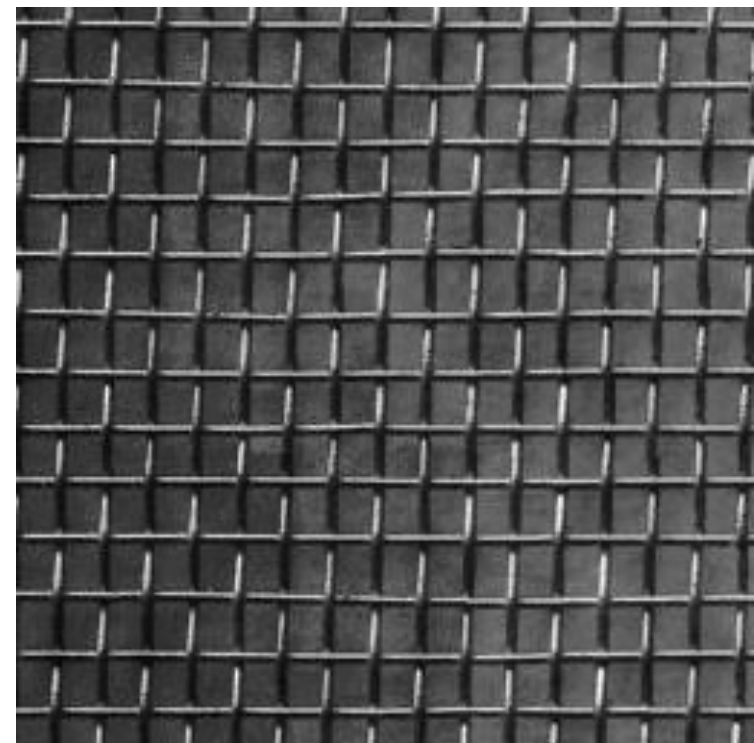
Portilla & Simoncelli



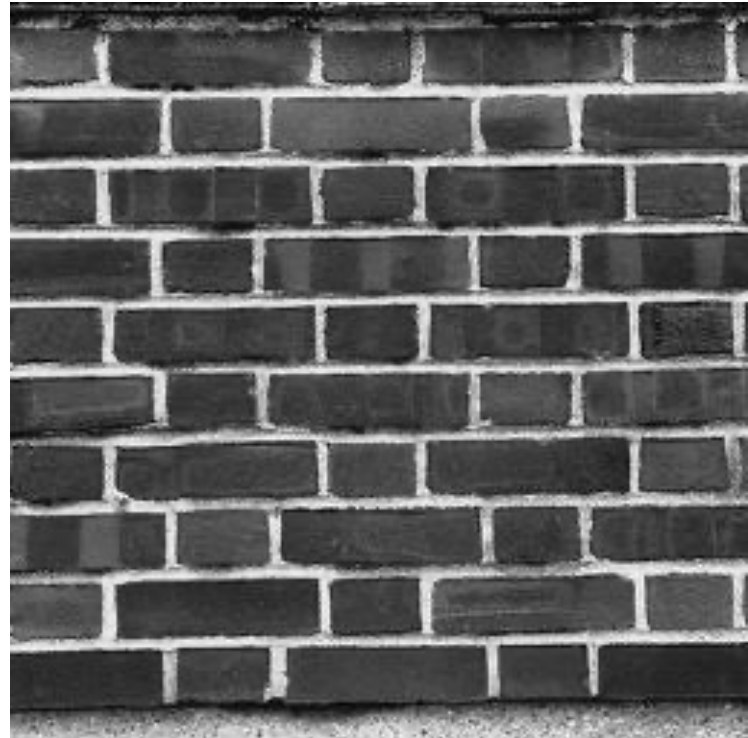
Xu, Guo & Shum



Wei & Levoy



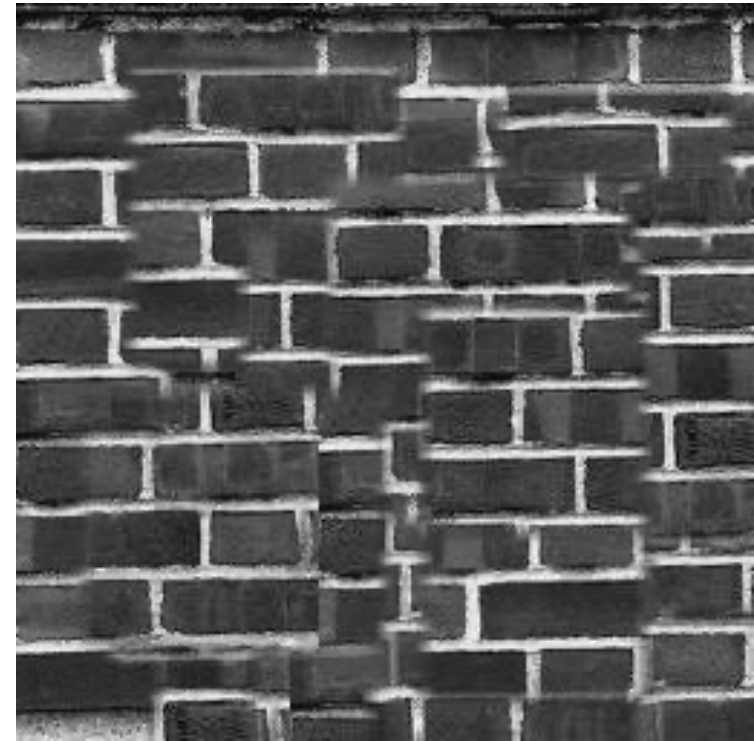
Efros and Freeman



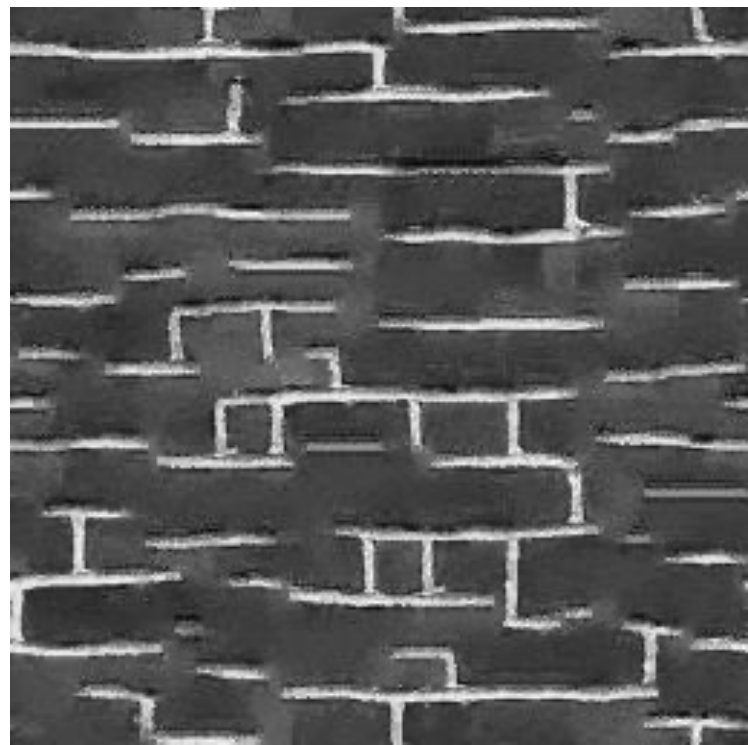
input image



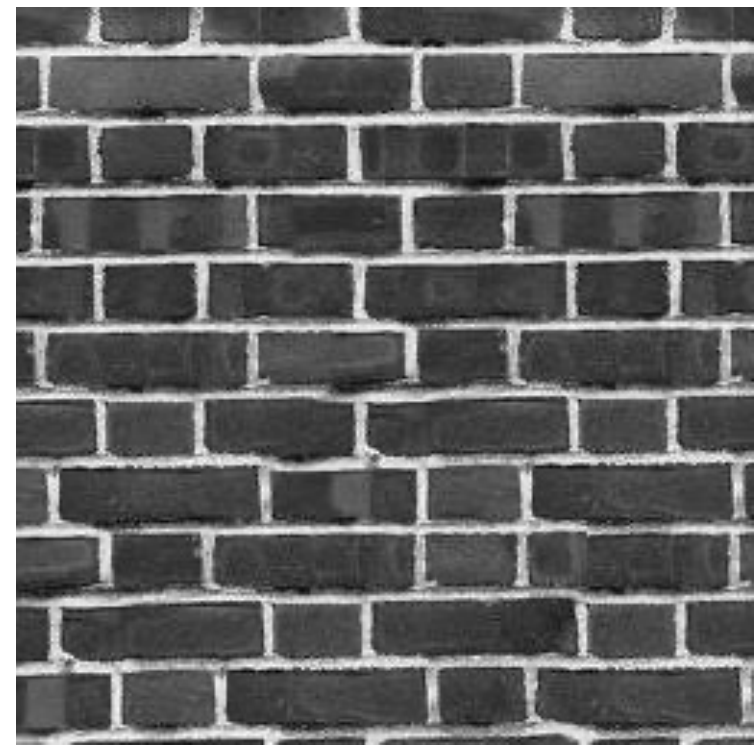
Portilla & Simoncelli



Xu, Guo & Shum



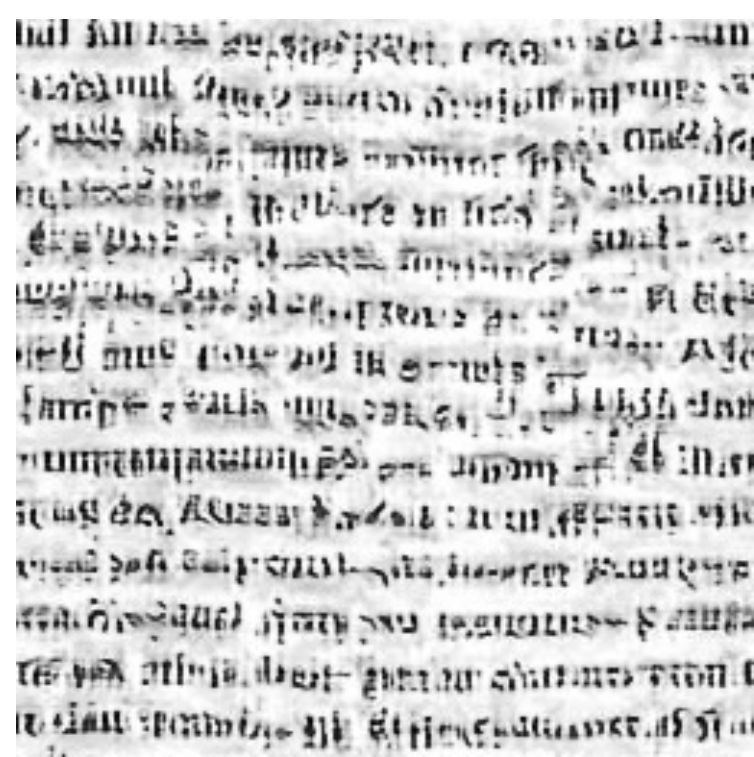
Wei & Levoy



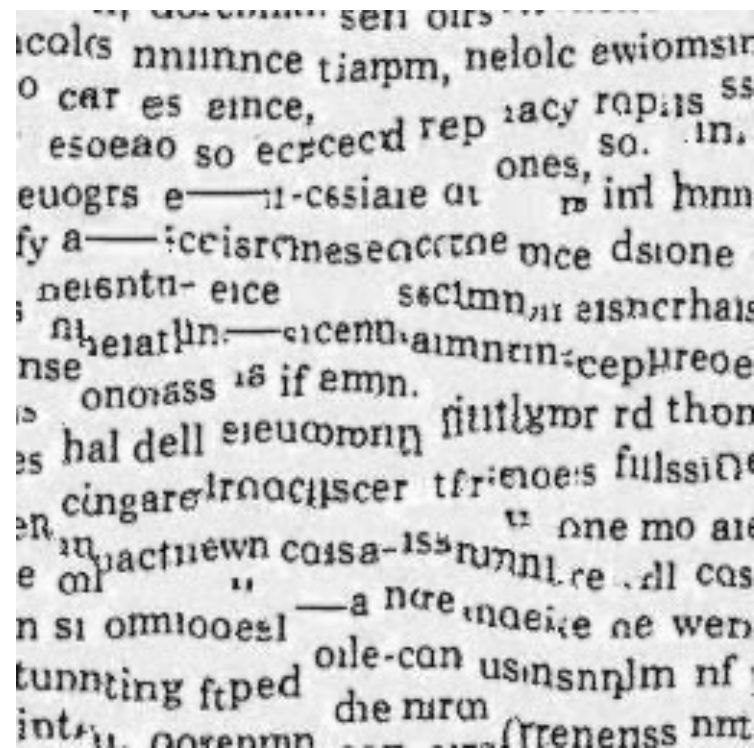
Efros and Freeman

... of a visual cortical neuron—the in
... describing the response of that neuro
... ht as a function of position—is perhap
... functional description of that neuron.
... seek a single conceptual and mathem
... describe the wealth of simple-cell recep
... and neurophysiologically¹⁻³ and inferred
... especially if such a framework has the
... it helps us to understand the functio
... leeper way. Whereas no generic mo
... ussians (DOG), difference of offset C
... rivative of a Gaussian, higher derivati
... function, and so on—can be expect
... simple-cell receptive field, we noneth

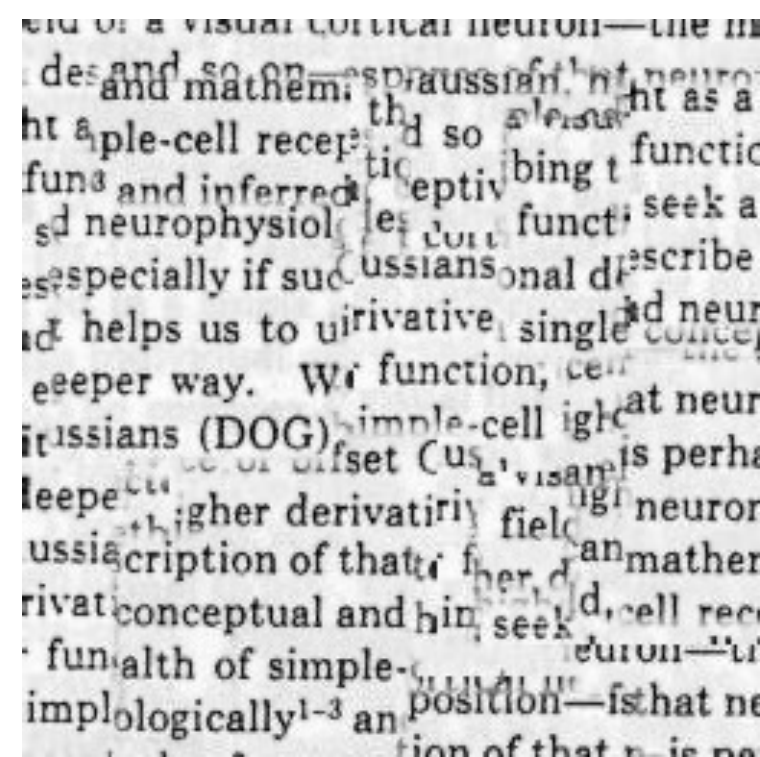
input image



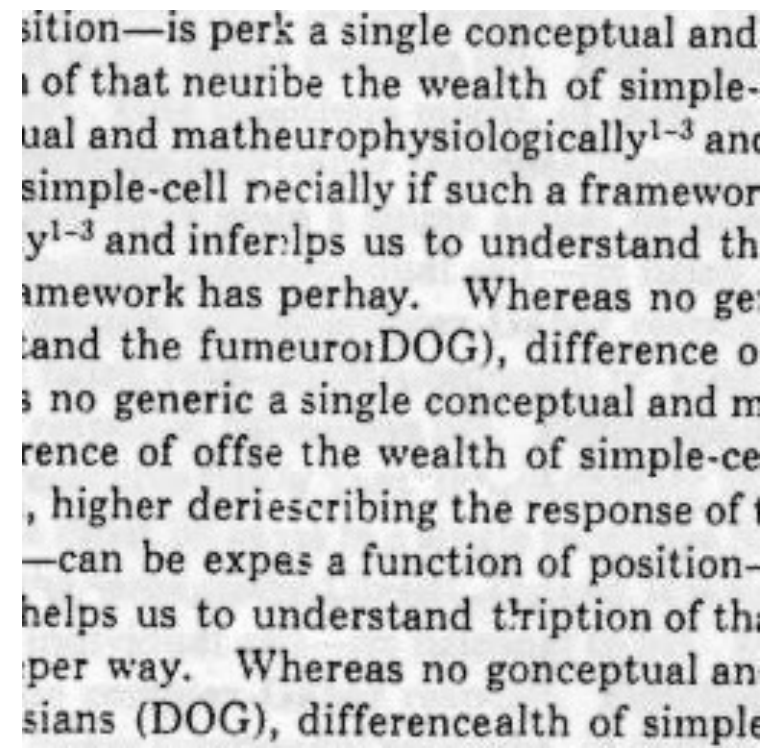
Portilla & Simoncelli



Wei & Levoy



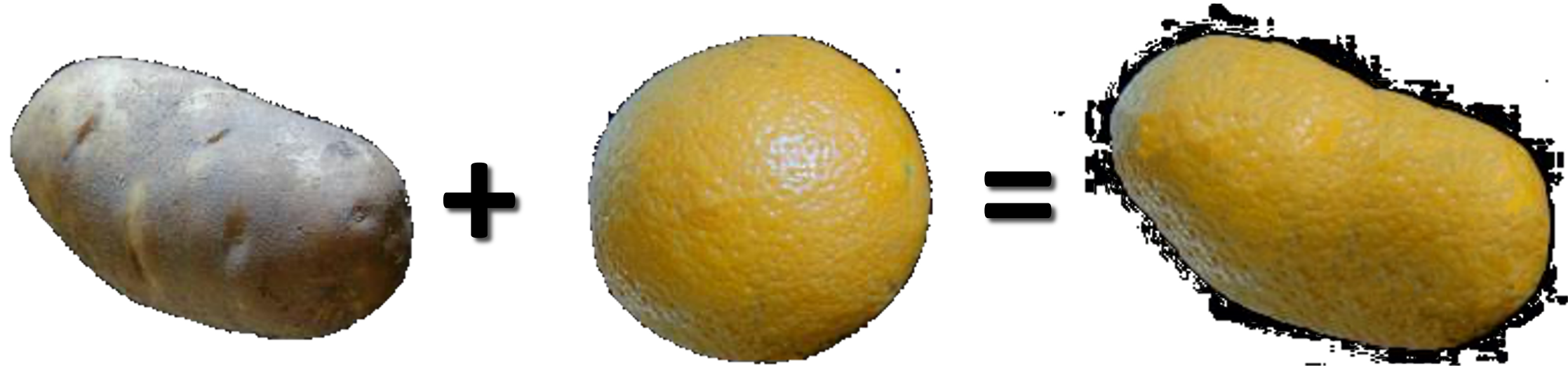
Xu, Guo & Shum



Efros and Freeman

Application: Texture Transfer

- Try to explain one object with bits and pieces of another object:

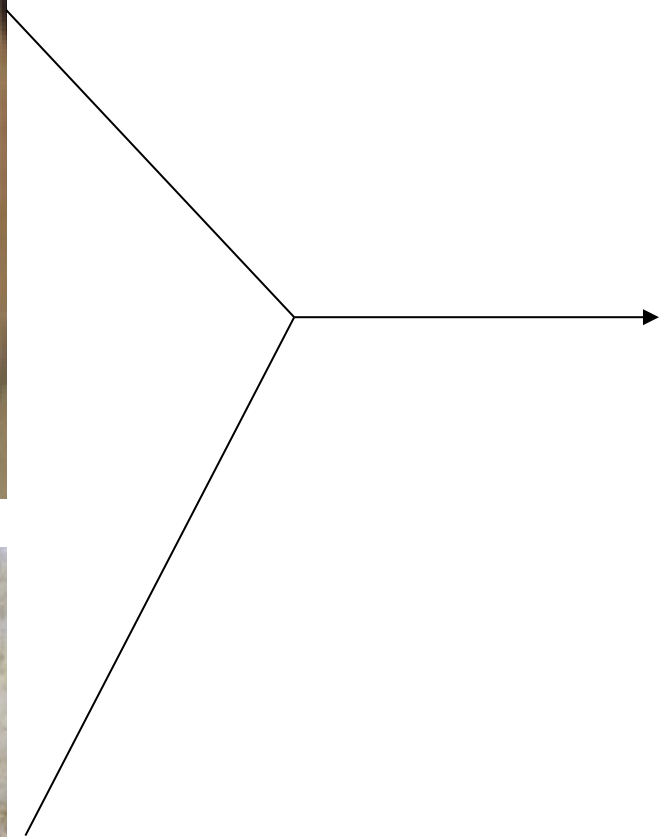


Texture Transfer

Constraint



Texture sample



Texture Transfer

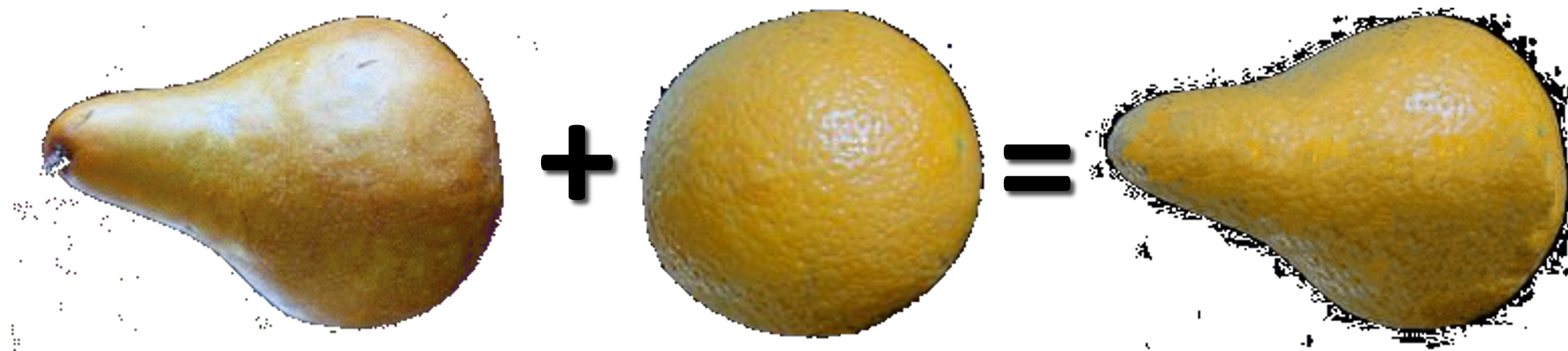
- Take the texture from one image and “paint” it onto another object



Same as texture synthesis, except an additional constraint:

1. Consistency of texture
2. Similarity to the image being “explained”

Texture Transfer



Texture Transfer

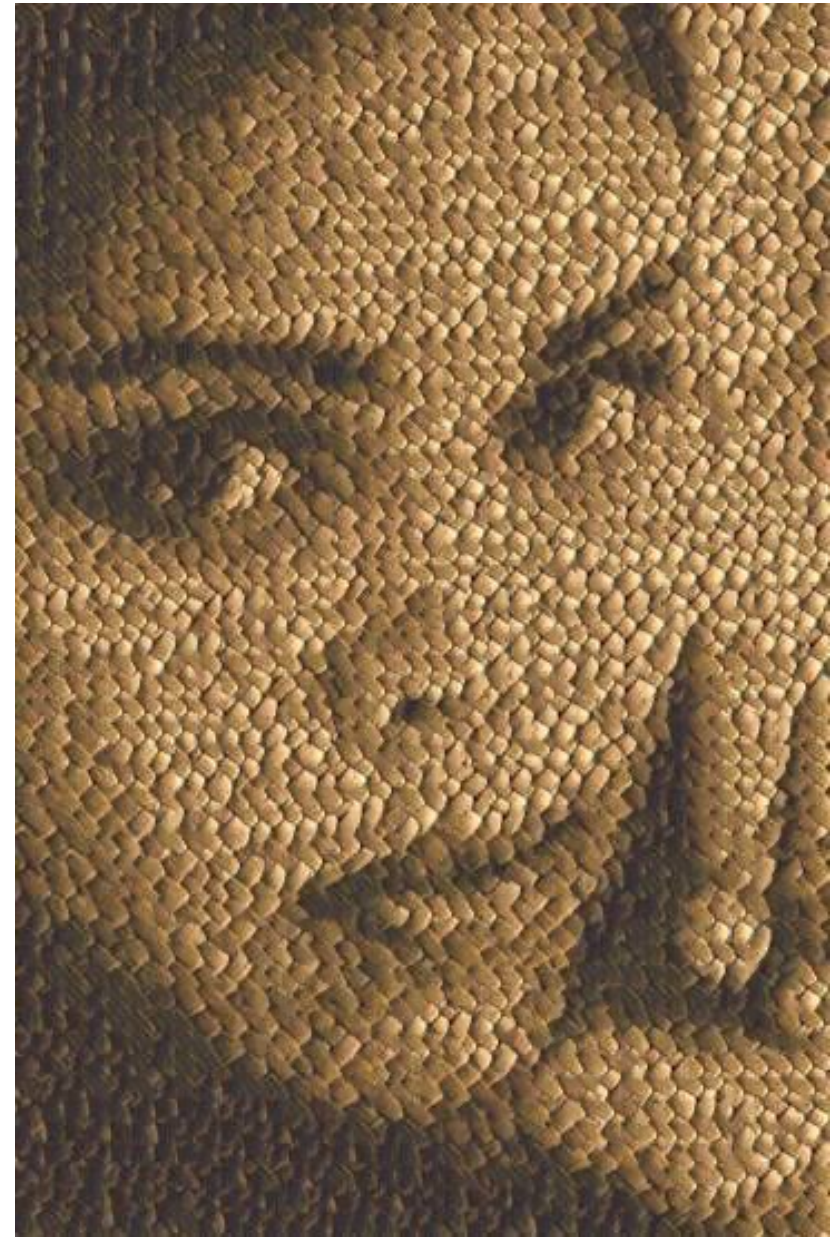


Image Analogies

Aaron Hertzmann^{1,2}

Chuck Jacobs²

Nuria Oliver²

Brian Curless³

David Salesin^{2,3}

¹**New York University**

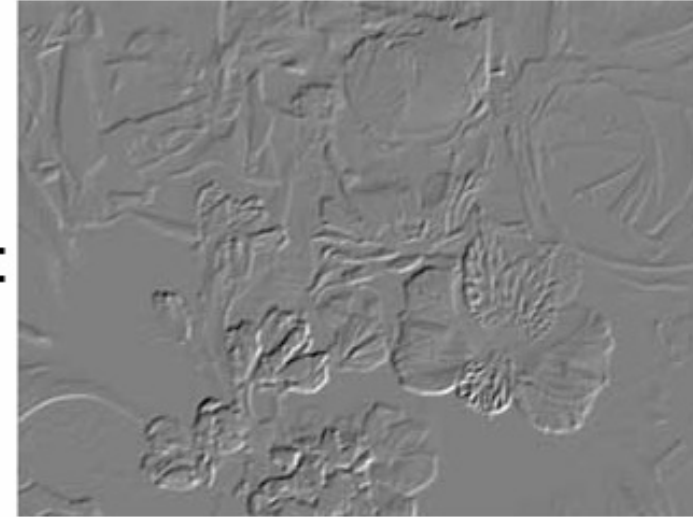
²**Microsoft Research**

³**University of Washington**

Edge Filter



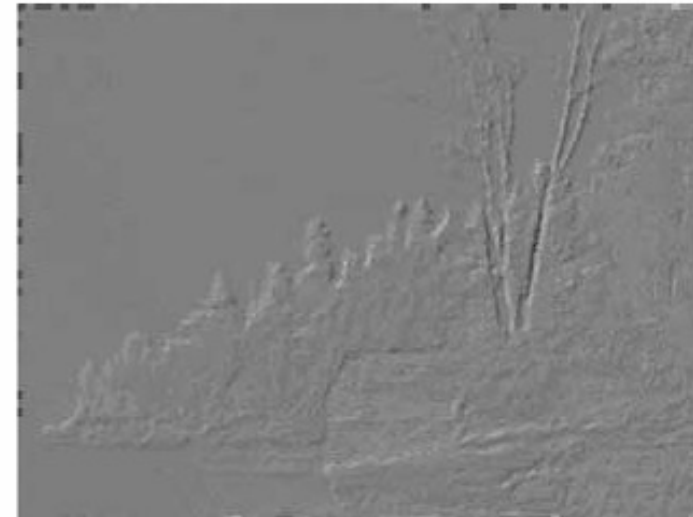
Unfiltered source (A)



Filtered source (A')



Unfiltered target (B)



Filtered target (B')

Artistic Filters



A



A'



B



B'

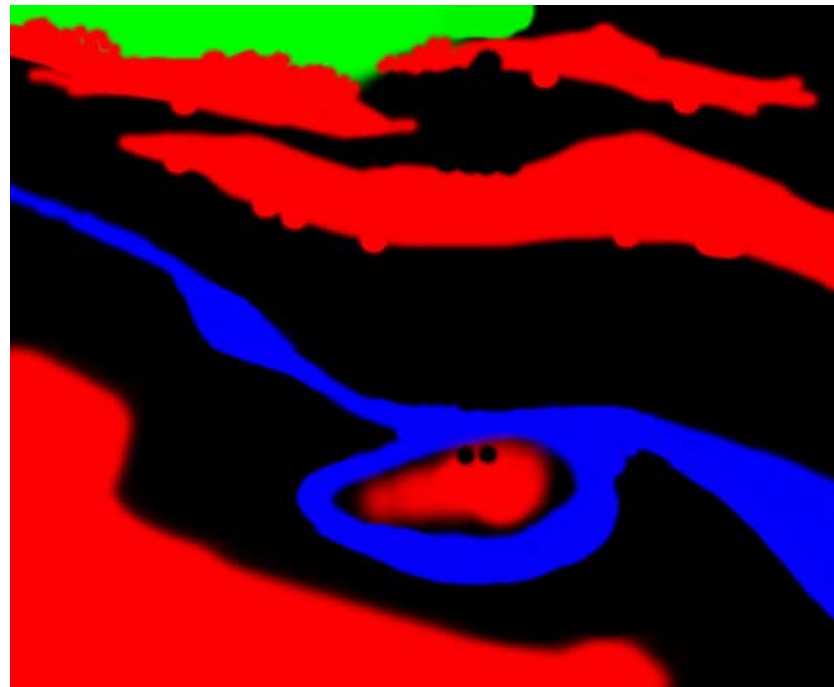
Texture-by-numbers



A



A'

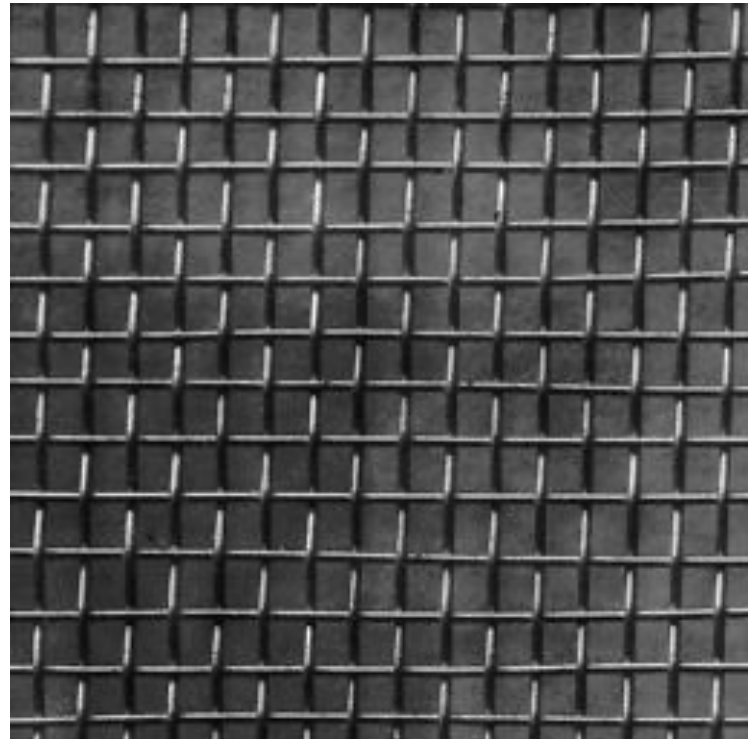


B

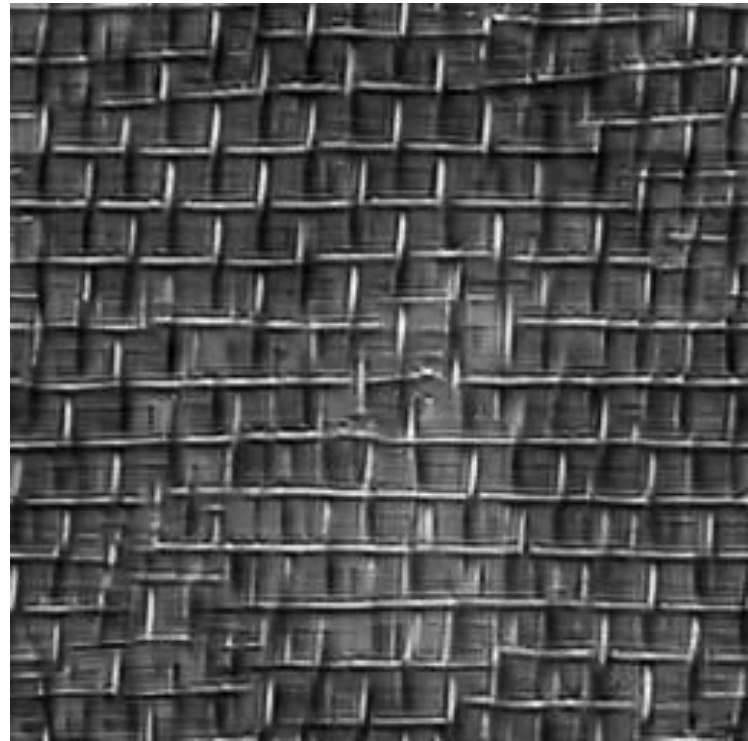


B'

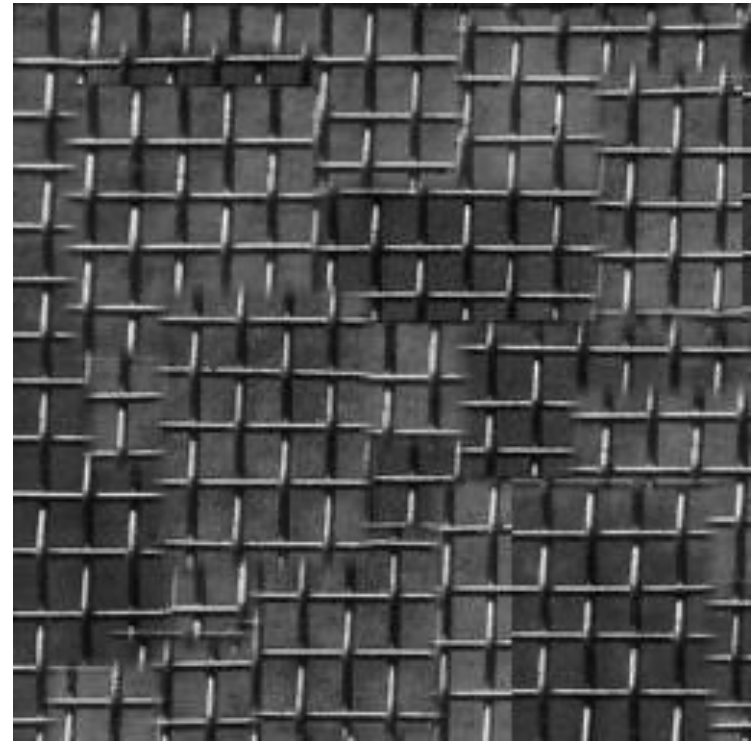
Parametric Texture Synthesis



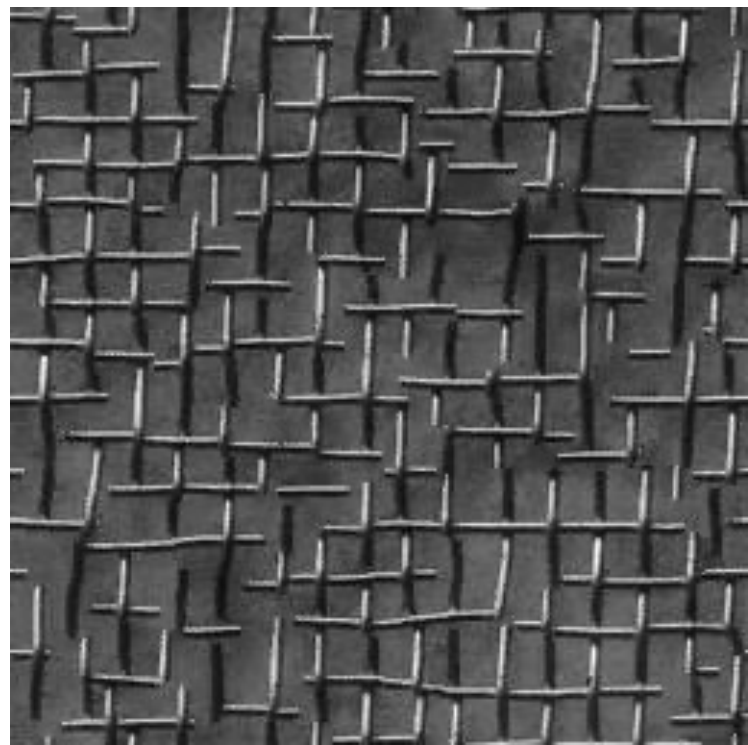
input image



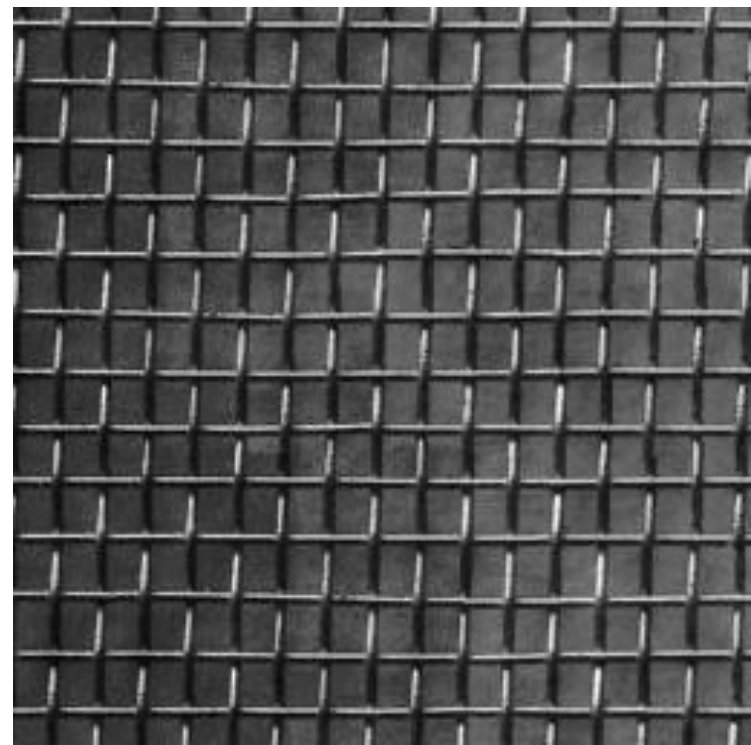
Portilla & Simoncelli



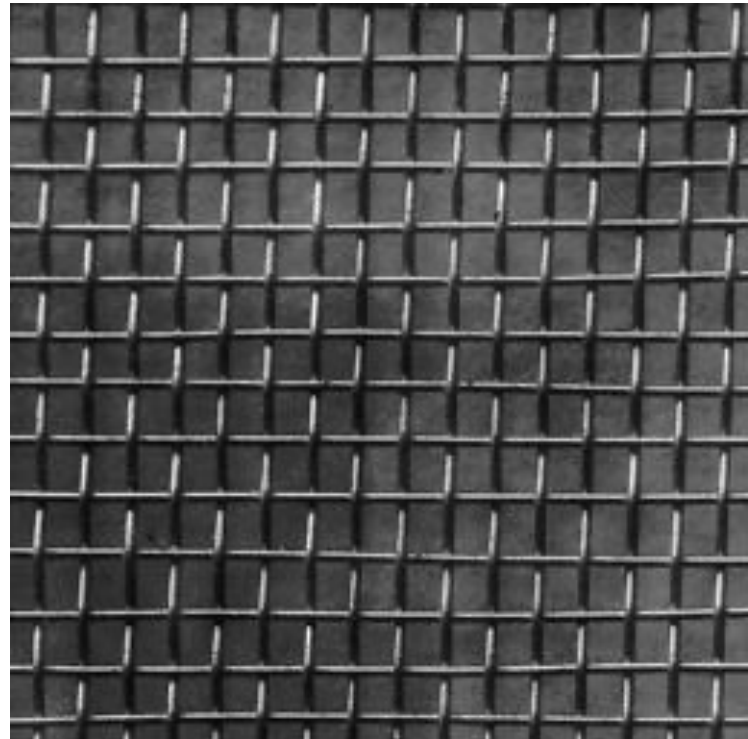
Xu, Guo & Shum



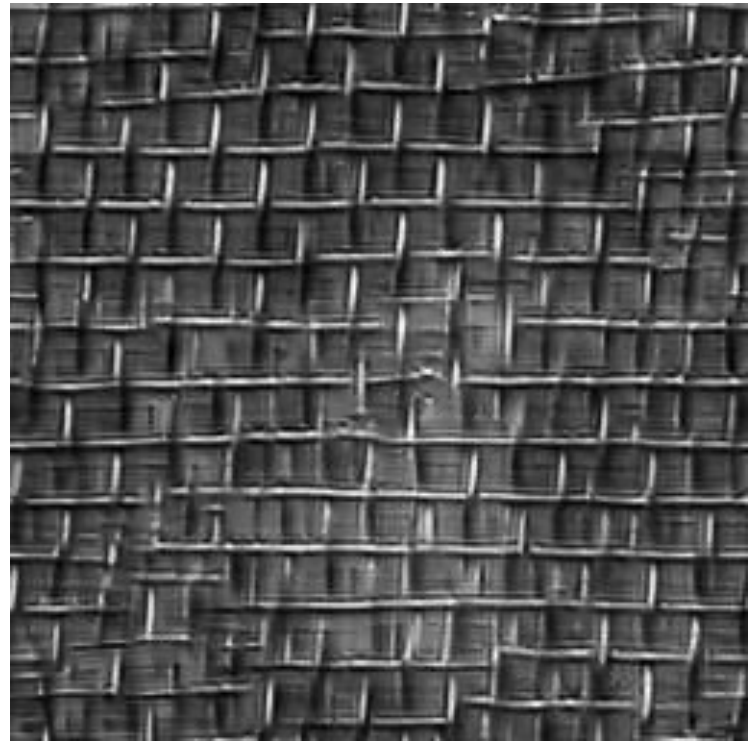
Wei & Levoy



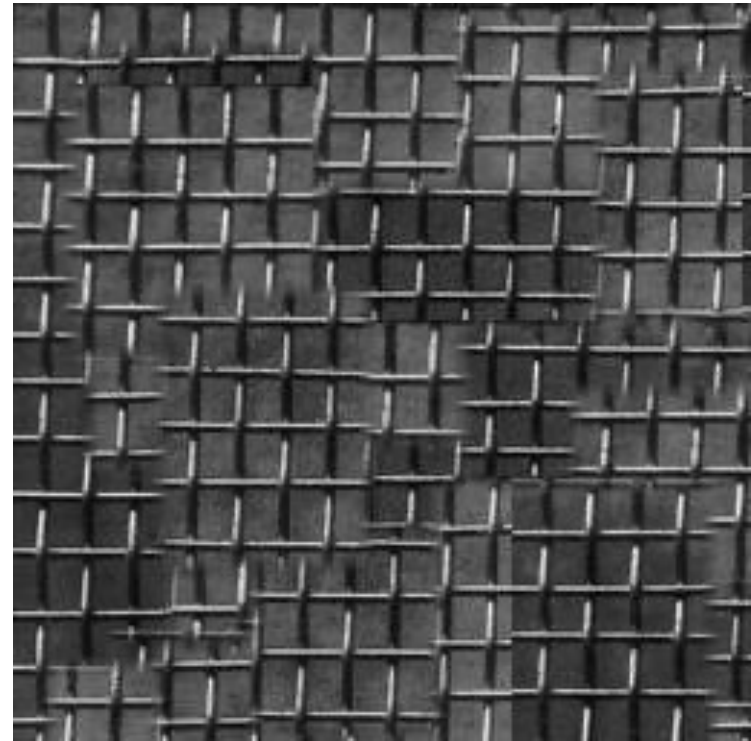
Efros and Freeman



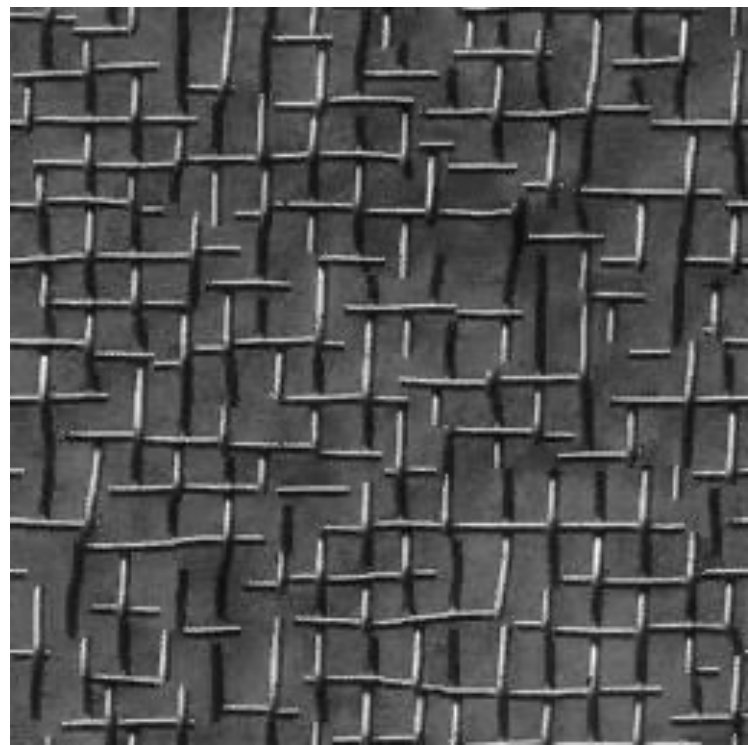
input image



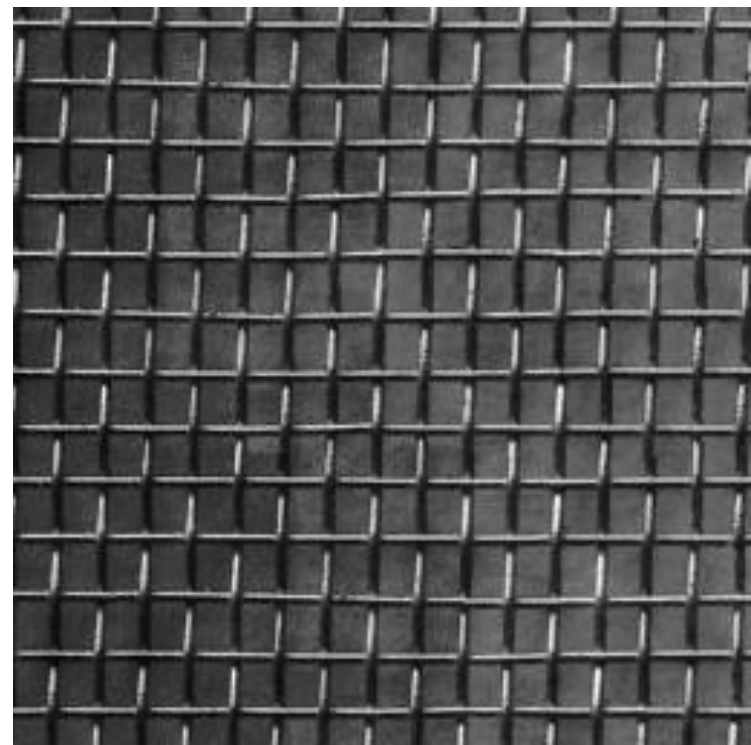
Portilla & Simoncelli



Xu, Guo & Shum

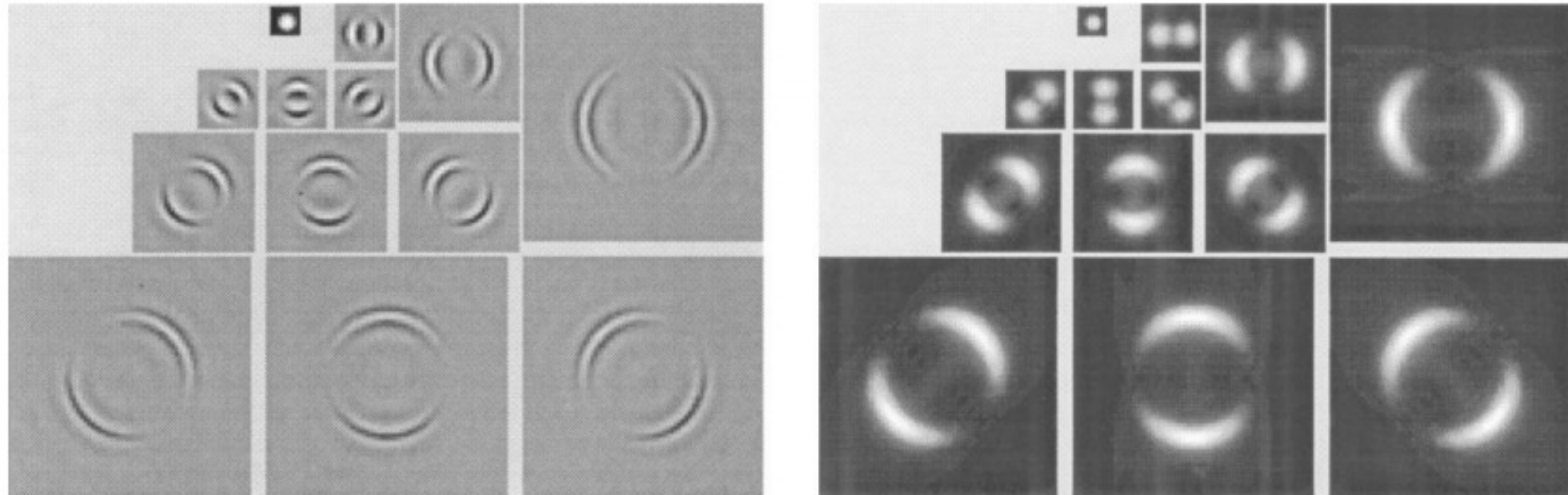


Wei & Levoy



Efros and Freeman

Parametric Texture Synthesis



Histogram and cross-channel correlation using wavelet basis

Statistics $\longrightarrow \mathcal{E}(\phi_j(y)) \approx \mathcal{E}(\phi_j(\hat{y}))$

Wavelet features

A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients

Portilla and Simoncelli, IJCV 1999

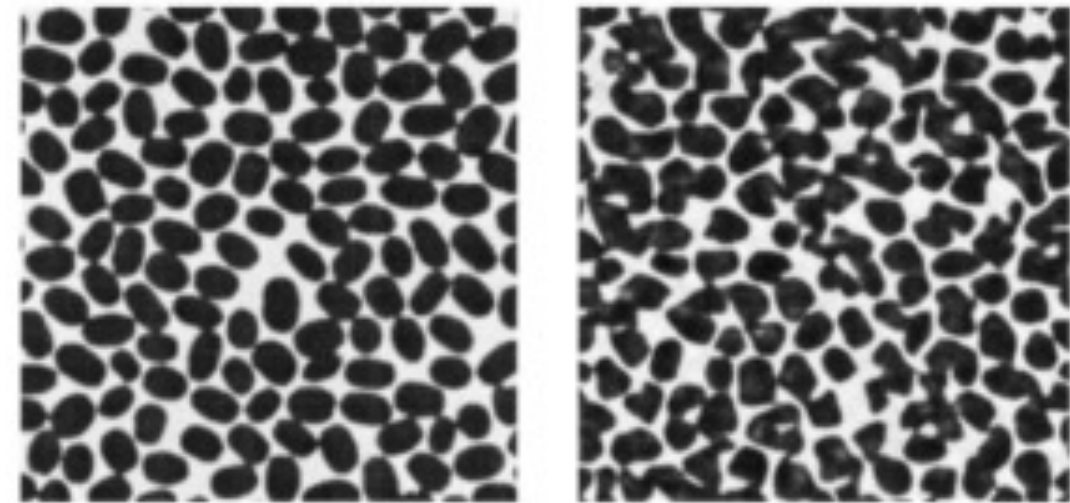
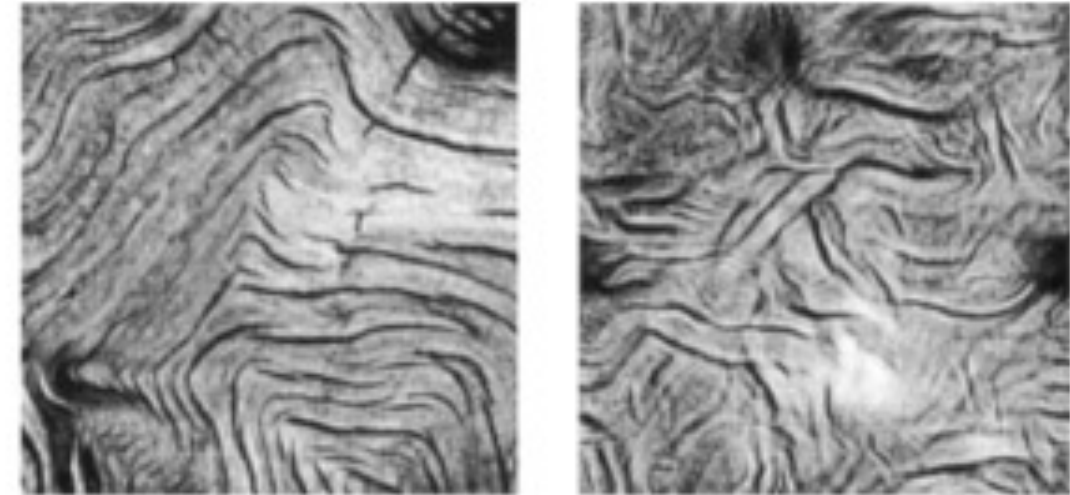
Parametric Texture Synthesis

Objective function

Given input texture y , feature descriptor ϕ ,
and statistics summary function \mathcal{E}

We aim to optimize the output image \hat{y}

$$\hat{y}^* = \arg \min_{\hat{y}} \|\mathcal{E}(\phi_j(\hat{y})) - \mathcal{E}(\phi_j(y))\|$$



Original y

Output \hat{y}

Deep Learning Version

Gram matrix:

- Cross Correlation of CNN features
- Invariant to the feature locations

$$V = [v_1, v_2, \dots, v_n]$$

$$G_{ij} = \langle v_i, v_j \rangle \quad G = V^T V$$

$$Gram^{(j)}(x) = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}$$

h, w: pixel locations index

c: channel index

H, W: height and width of feature map

C: the number of total channels

Style Reconstruction (Style Loss)

$$\left| \text{Gram} \left(\begin{array}{c} \hat{y} \\ \text{optimized output} \end{array} \right) - \text{Gram} \left(\begin{array}{c} y \\ \text{style image} \end{array} \right) \right|$$

Gram = Gram Matrix of a deep network's features (e.g., ImageNet classifier)

Style Loss

$$\arg \min_{\hat{y}} \sum_j^M \lambda_j \left\| \text{Gram}^{(j)}(\hat{y}) - \text{Gram}^{(j)}(y) \right\|^2$$

weight λ_j (j)-th layer

Portilla & Simoncelli



original



pool4



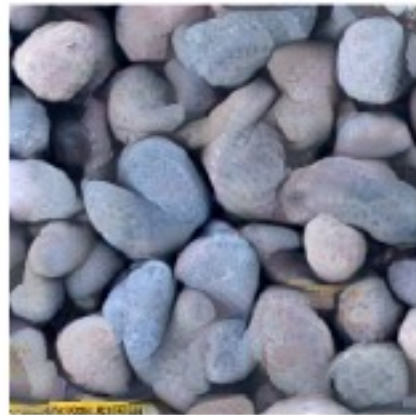
pool3



A ~1k parameters



~10k parameters



~177k parameters



~852k parameters

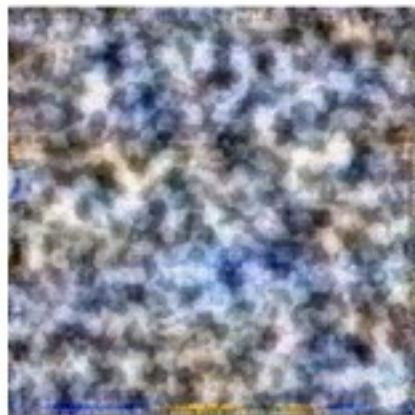


original



Number of parameters

B conv1



conv2



conv3



conv4



conv5

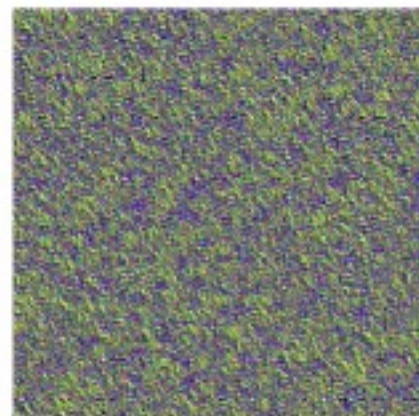


Different layers

C conv1_1



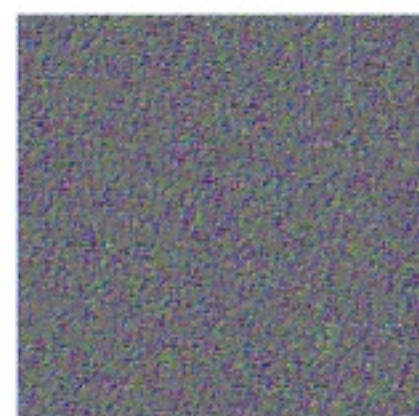
pool1



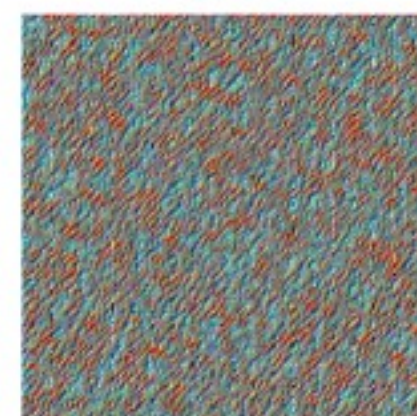
pool2



pool3



pool4



The same network architecture with random weights

Neural Style Transfer



content image

+



style image

=



output result

Content Reconstruction (Perceptual Loss)

$$\left| F\left(\overset{\hat{y}}{\text{optimized output}}\right) - F\left(\overset{x}{\text{content image}}\right) \right|$$

F is a deep network (e.g., ImageNet classifier)

Content Loss

$$\arg \min_{\hat{y}} \sum_i^N \overset{\text{weight}}{\lambda_i} \left\| F^{(i)}(\hat{y}) - F^{(i)}(x) \right\|_1$$

(i)-th layer

Content Reconstruction (Perceptual Loss)



Conv1_2

Conv2_2

Conv3_2

Conv4_2

Conv5_2

Neural Style Transfer

$$\left| \text{Gram}(\hat{y}) - \text{Gram}(y) \right|$$

optimized output style image

$$+ \left| F(\hat{y}) - F(x) \right|$$

optimized output content image

$$\arg \min_{\hat{y}} \mathcal{L}_{\text{style}}(\hat{y}, y) + \lambda \mathcal{L}_{\text{content}}(\hat{y}, x)$$



Different Initializations



Fast Neural Style Transfer

- Optimization-based method

$$\arg \min_{\hat{y}} \mathcal{L}_{\text{style}}(\hat{y}, y) + \lambda \mathcal{L}_{\text{content}}(\hat{y}, x)$$

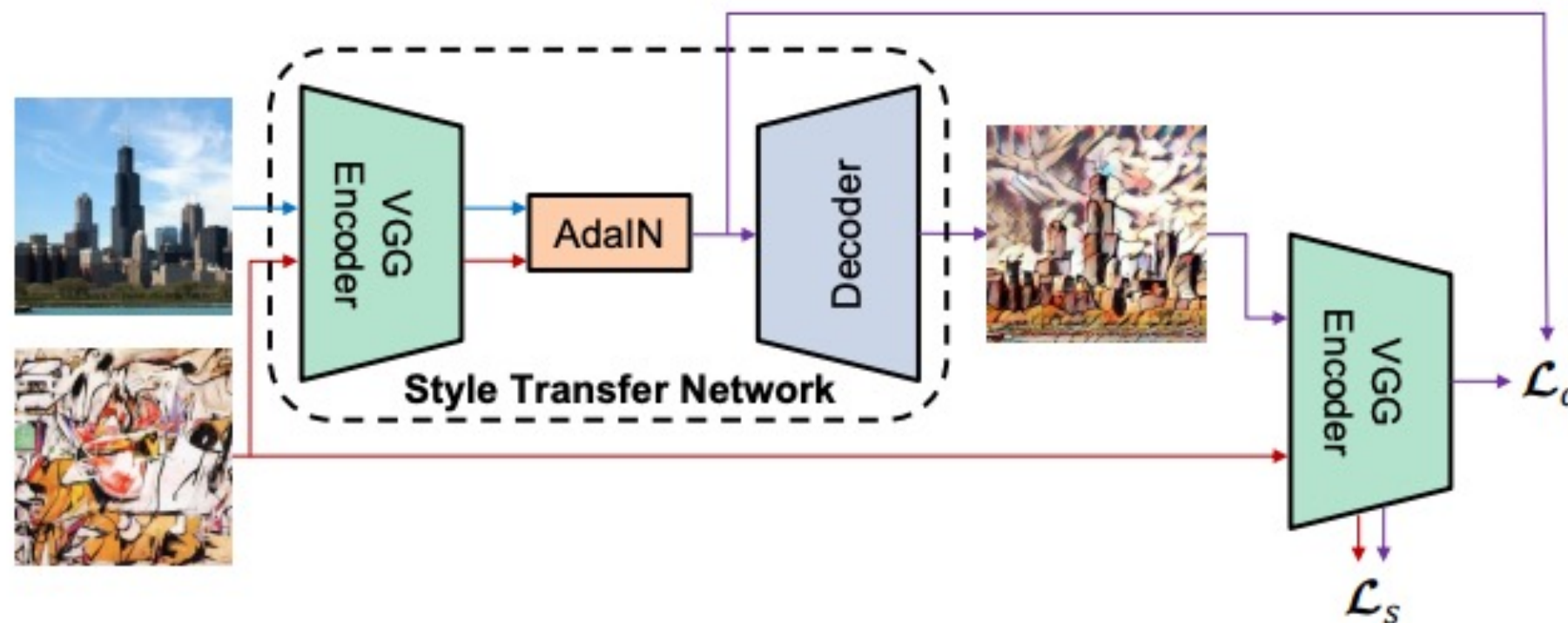
- Feedforward network

$$\arg \min_G \mathbb{E}_x \mathcal{L}_{\text{style}}(G(x), y) + \lambda \mathcal{L}_{\text{content}}(G(x), x)$$

Arbitrary Style Transfer with AdaIN

- Feedforward network with any style

$$\arg \min_G \mathbb{E}_{x,y} \mathcal{L}_{\text{style}}(G(x, y), y) + \lambda \mathcal{L}_{\text{content}}(G(x, y), x)$$

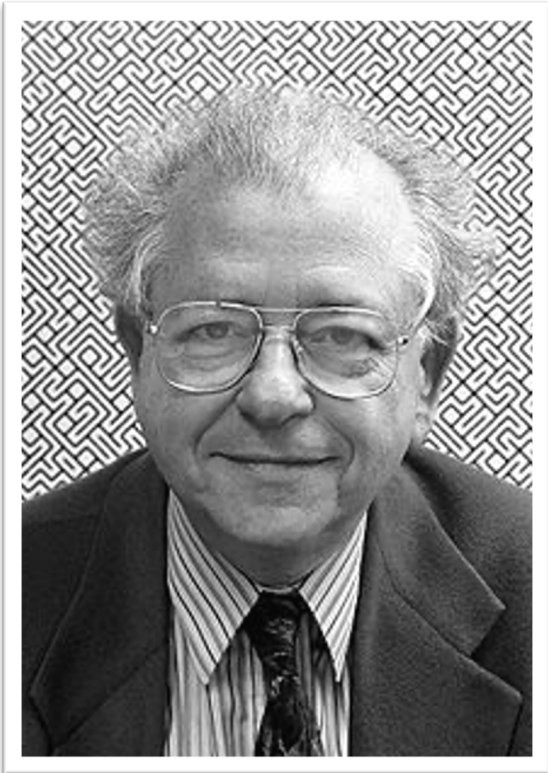


Arbitrary Style Transfer with AdaIN

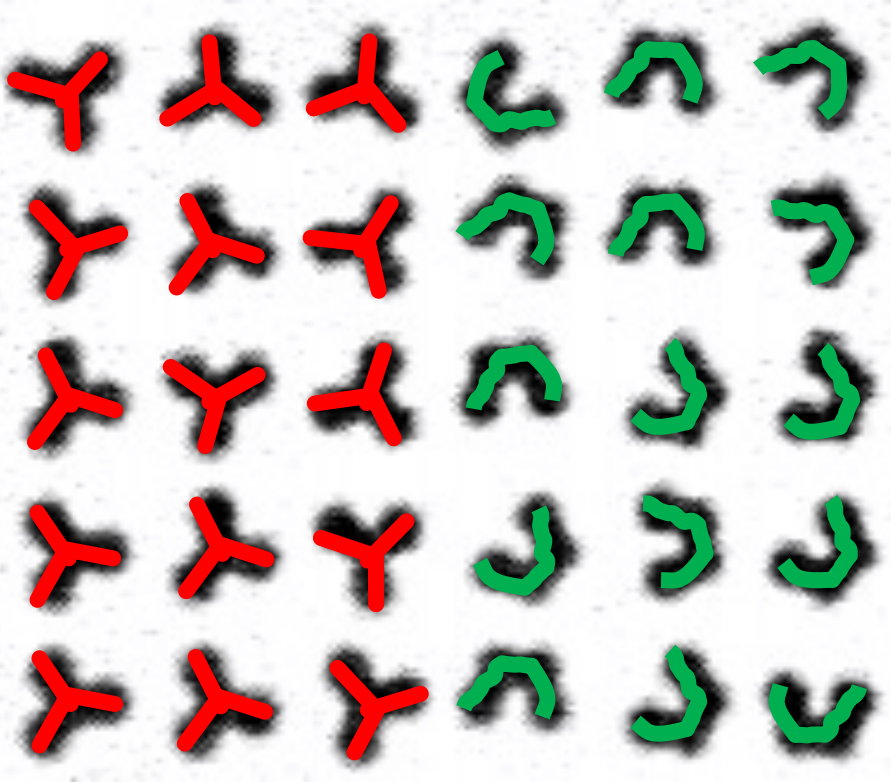


Texture Discrimination

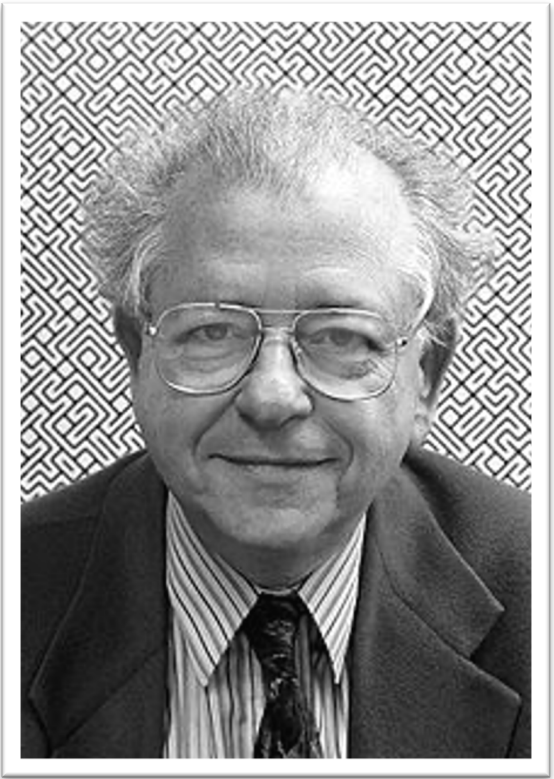
Texture Discrimination in Human Perception



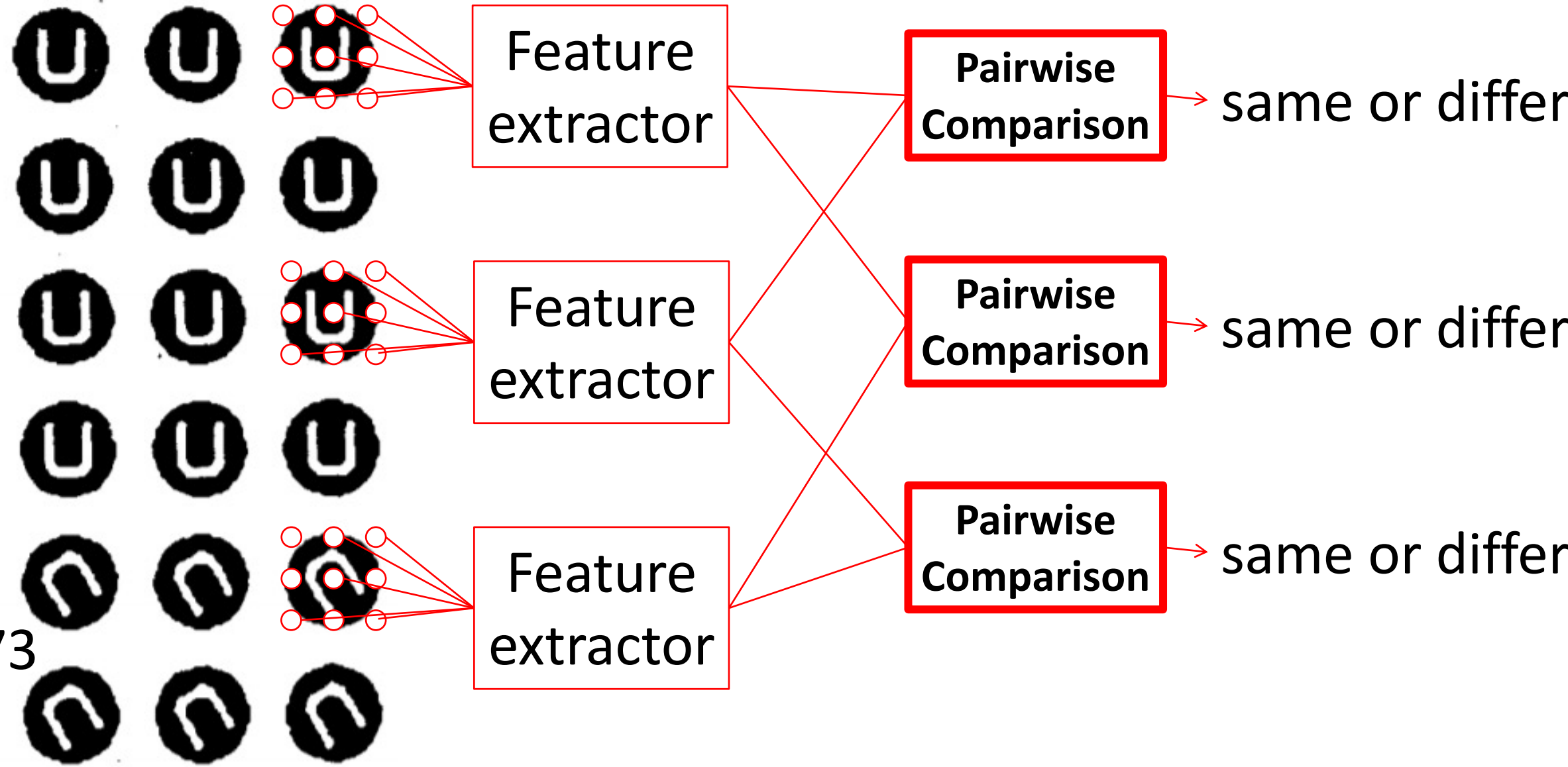
Bela Julesz, 1973



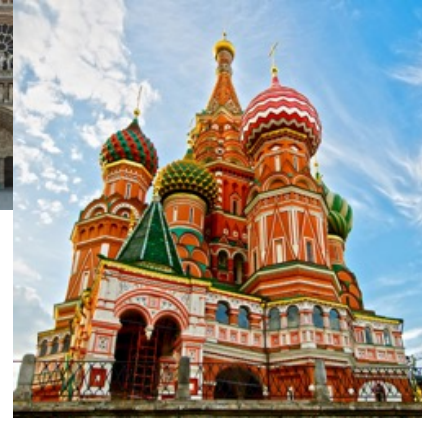
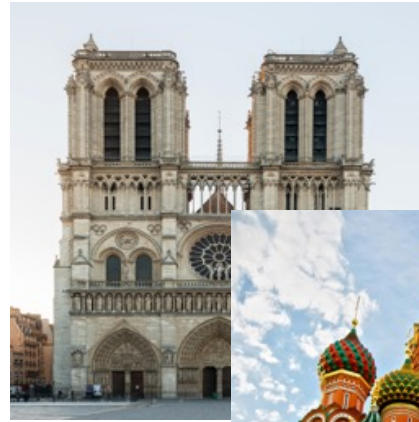
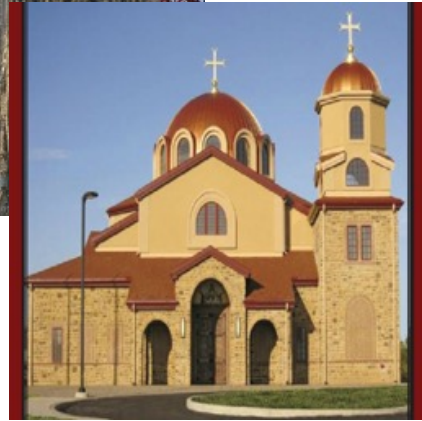
Texture Discrimination in Human Perception



Bela Julesz, 1973



Church images



Are



and



from the same image?



Answer:
No

Are



and

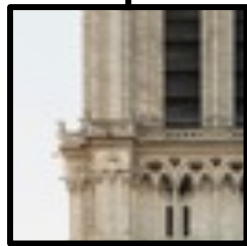


from the same image?



Answer:
Yes

Are



and



from the same image?

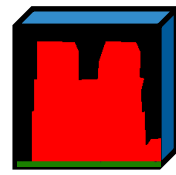
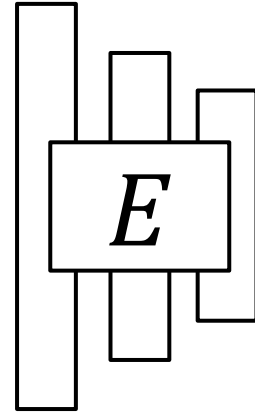


Answer:
...?

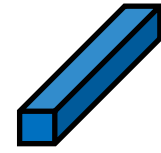
Are  and  from the same image?

Patch co-occurrence discriminator

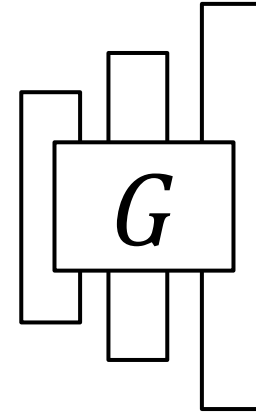
Auto-
encode



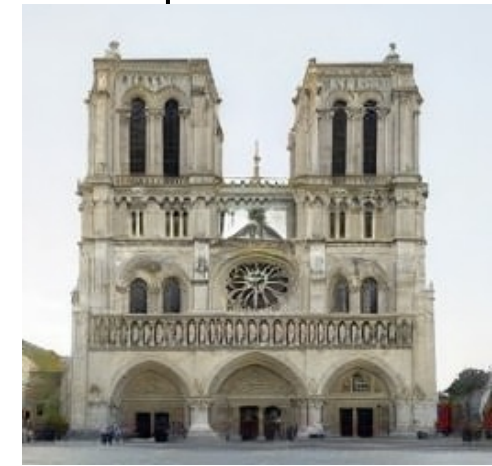
structure code



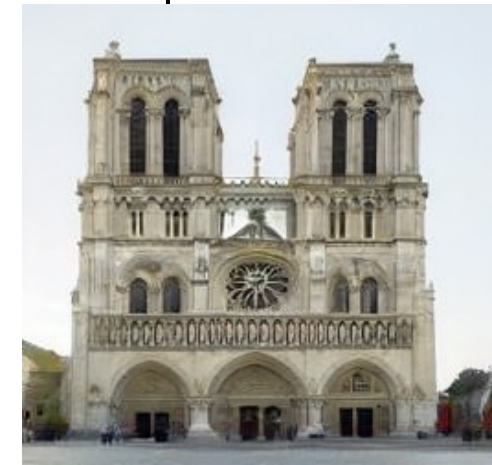
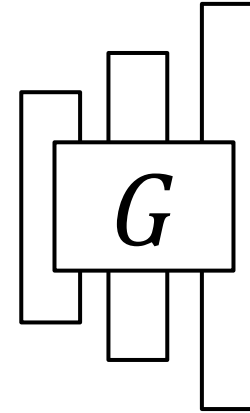
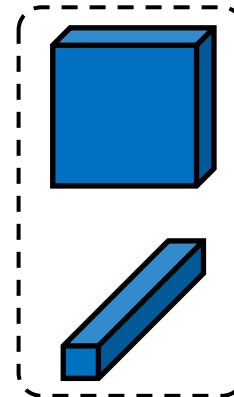
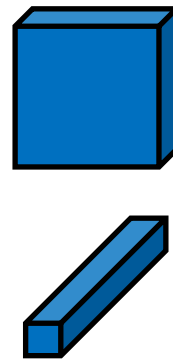
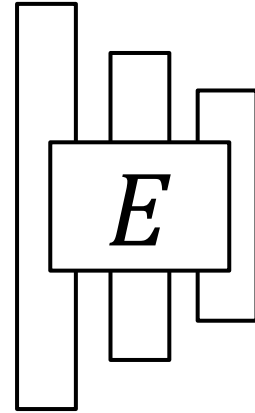
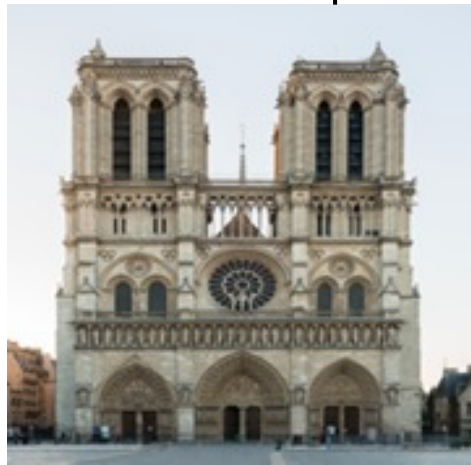
texture code



Reconstruction

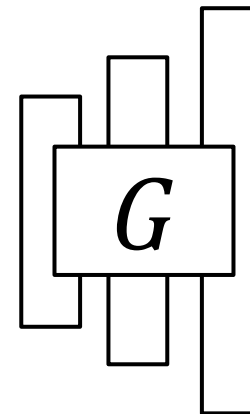
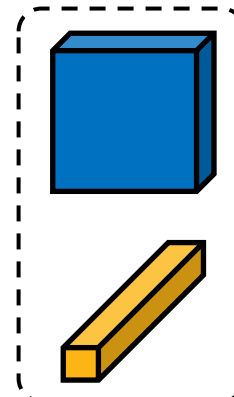
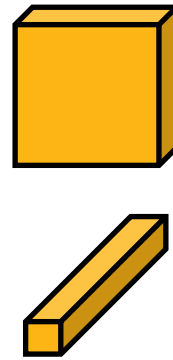
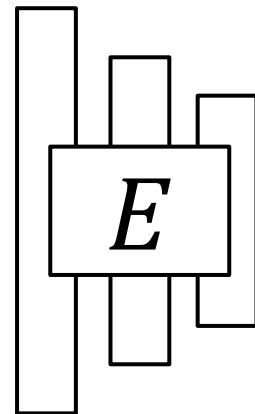
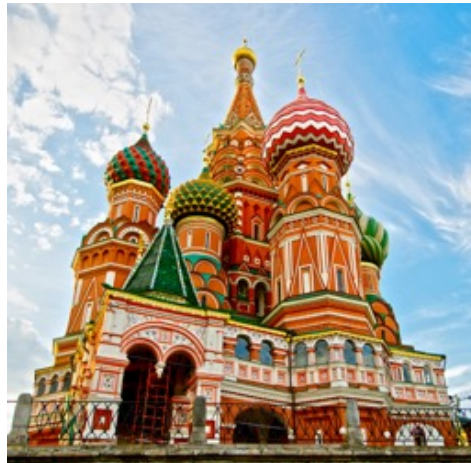


Auto-
encode

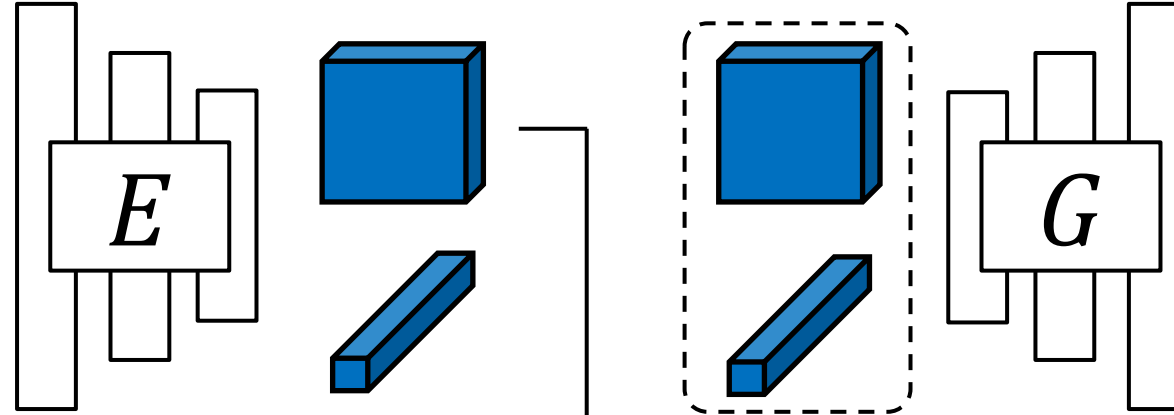
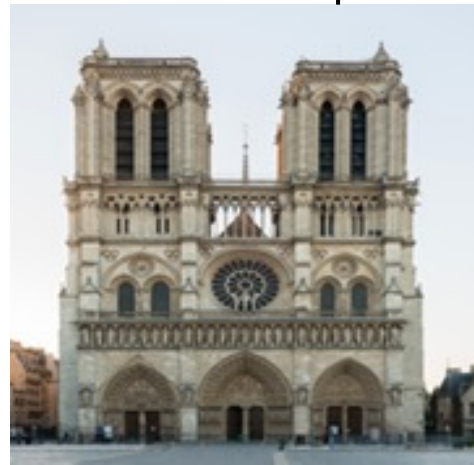


Reconstruction

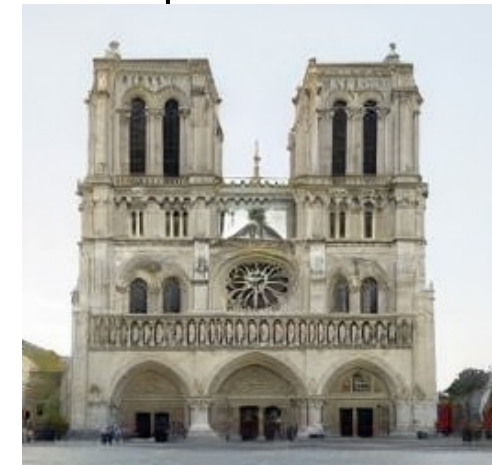
Swap



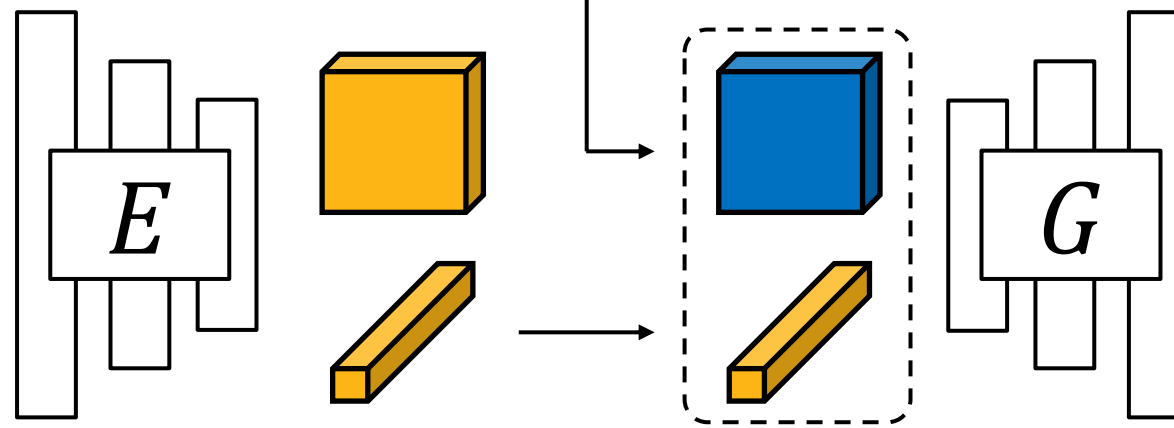
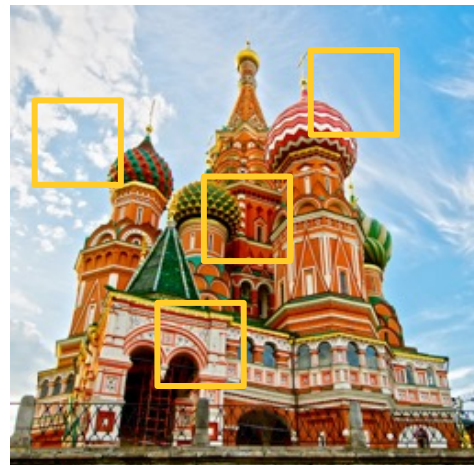
Auto-
encode



Reconstruction



Swap



texture

structure

