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

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
Can you match the texture to the response?

Filters

1

2

3

Mean abs. responses

A

B

C

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
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

Unit activations

0.1  
0.2  
0.5  
0.0  
0.2  

dog  
cat  
gecko  
alligator  
fish
Extracting neural net features

$c_i(x, y)$

112 width

112 height

128 channels

conv2 feature activations

$(112 \times 112) \times 128$ for conv2 of VGG19
Capturing feature correlations

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

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.
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.

Perceptual loss:
usually distance in feature space

... and the **content** of the photo.
London during the day.

New York at night.
Neural networks that generate images
Image classification

: image $\mathbf{x}$

Classifier

"Duck"

label $y$

Source: Isola, Freeman, Torralba
Image synthesis

“Duck” \rightarrow \text{Generator} \rightarrow \text{image } x

label y

Source: Isola, Freeman, Torralba
Neural networks as distribution transformers

Source distribution

$\mathcal{G} \quad p(z)$

Target distribution

$\mathcal{G} \quad p(x)$

Source: Isola, Freeman, Torralba
Neural networks as distribution transformers

\[ z \sim \mathcal{N}(0, 1) \]

Gaussian noise

Synthesized image

Source: Isola, Freeman, Torralba
Gaussian noise $z \sim \mathcal{N}(\mathbf{0}, 1)$

Neural networks as distribution transformers

Source: Isola, Freeman, Torralba
Generative adversarial networks (GANs)
\( G \) tries to synthesize fake images that fool \( D \)

\( D \) tries to identify the fakes

Source: Isola, Freeman, Torralba

[Goodfellow et al., 2014]
$\arg\max_D \mathbb{E}_{z,x} \left[ \log D(G(z)) + \log (1 - D(x)) \right]$
\(G\) tries to synthesize fake images that \textit{fool} \(D\):

\[
\arg\min_G \mathbb{E}_{z,x} \left[ \log D(G(z)) + \log (1 - D(x)) \right]
\]

Source: Isola, Freeman, Torralba

[Goodfellow et al., 2014]
\( \textbf{G} \) tries to synthesize fake images that \textit{fool} the \textit{best} \( \textbf{D} \):

\[
\arg \min_G \max_D \mathbb{E}_{z,x} \left[ \log D(G(z)) + \log (1 - D(x)) \right]
\]

Source: Isola, Freeman, Torralba

[Goodfellow et al., 2014]
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]


Source: Isola, Freeman, Torralba
Latent space (Gaussian) \( \mathbf{z} \)  
Data space (Natural image manifold) \( \mathbf{X} \)

[BigGAN, Brock et al. 2018]
StyleGAN2
https://github.com/NVlabs/stylegan3

StyleGAN3 (Ours)

[Karras et al., “Alias-Free Generative Adversarial Networks”, 2021]
[Karras et al., “Alias-Free Generative Adversarial Networks”, 2021]
StyleGAN2

StyleGAN3 (Ours)

[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 → Translator → Satellite photo

Source: Isola, Freeman, Torralba
Map2Sat

\[
\begin{align*}
\{ \text{x} \} & , \{ \text{y} \} \\
\{ \text{x} \} & , \{ \text{y} \} \\
\end{align*}
\]
Idea: L1 loss

$$|| G(x) - y ||_1$$
\textbf{G} tries to synthesize fake images that fool \textbf{D}

\textbf{D} tries to identify the fakes
\[
\text{arg max}_D \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right]
\]

Source: Isola, Freeman, Torralba
\( \min G \) 

\[ \text{tries to synthesize fake images that fool } D: \]

\[ \arg\min_G \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right] \]

Source: Isola, Freeman, Torralba
\( \mathbf{G} \) tries to synthesize fake images that fool the best \( \mathbf{D} \):

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

Source: Isola, Freeman, Torralba
\[
\arg \min_G \max_D \mathbb{E}_{x,y} \left[ \log D(G(x)) + \log(1 - D(y)) \right]
\]

**Source:** Isola, Freeman, Torralba
\[
\arg\min_G \max_D \mathbb{E}_{x, y}[\log D(G(x)) + \log(1 - D(y))]
\]

Source: Isola, Freeman, Torralba
\[
\arg \min_{G} \max_{D} \mathbb{E}_{x, y}[ \log D(G(x)) + \log(1 - D(y)) ]
\]

Source: Isola, Freeman, Torralba
arg min \max_D \mathbb{E}_{x,y} \left[ \log D(x, G(x)) + \log(1 - D(x, y)) \right]
\[
\arg \min_G \max_D \mathbb{E}_{x,y} [ \log D(x, G(x)) + \log(1 - D(x, y)) ]
\]
\[
\arg\min_G \max_D \mathbb{E}_{x,y} \left[ \log D(x, G(x)) + \log (1 - D(x, y)) \right]
\]
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
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

\[ \begin{align*}
\mathbf{x} & \quad \mathbf{y} \\
\{ & \quad \}
\end{align*} \]

[Source: Isola, Freeman, Torralba, 2015]
edges2cats

[Chris Hess, edges2cats]
Architectures

Discriminator: fully convolutional network

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

Sequence of strided convolutions

real or fake?
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

512x512x3  512x512x64
More recent architectures

[“GauGAN”, Park et al., CVPR 2019]
Handling unpaired data

Paired data

\[ x_i \quad y_i \]

Unpaired data

\[ X \quad Y \]

Source: Isola, Freeman, Torralba
arg min \max_G \mathbb{E}_{x,y} \left[ \log D(x, G(x)) + \log(1 - D(x, y)) \right]
arg min \max_{G} \mathbb{E}_{x,y} [ \log D(x, G(x)) + \log(1 - D(x, y)) ]

No input-output pairs!

Source: Isola, Freeman, Torralba
Real!
Real too!

Nothing to force output to correspond to input

Source: Isola, Freeman, Torralba
CycleGAN

Source: Isola, Freeman, Torralba
Cycle Consistency Loss

\[
\begin{align*}
X & \xrightarrow{G} \hat{Y} \quad & G(X) & \xrightarrow{F} \hat{X} \\
\hat{Y} & \xrightarrow{F} \hat{X} \\
D_Y & \\
\|F(G(x)) - x\|_1
\end{align*}
\]

Source: Isola, Freeman, Torralba
Cycle Consistency Loss

Source: Isola, Freeman, Torralba

\[ \|F(G(x)) - x\|_1 \]
Next time: more image synthesis