Neural Style Transfer
Seminar in Computer Vision, Spring 2017
Itamar Talmi

Outline
- Texture Synthesis
- Feature Visualization
- Feature Inversion
- Style Transfer

Neural Style Transfer

Texture Synthesis
Given a sample patch of some texture, can we generate a bigger image of the same texture?

Input

Output
Texture Synthesis

Wei and Levoy, "Fast Texture Synthesis using Tree-structured Vector Quantization", SIGGRAPH 2000
Efros and Leung, "Texture Synthesis by Non-parametric Sampling", ICCV 1999

Style transfer – Image analogies

Style transfer – Image analogies

Summary:
- Transferring high-frequency texture information

Limitations:
- Use only low-level features to capture texture
- Needs both filtered and unfiltered images
- Main edges and colors stay the same

Style transfer

Improve ‘Image analogies’ algorithm by additionally informing the content transfer with edge orientation information

Ideal style transfer algorithm

• Extracts semantics from a content image
  • Trees, houses, river.....

• Learn texture from a texture\style image

“Ideal” style transfer algorithm

• Extracts semantics from a content image

• Learn texture from a texture\style image

• Renders the same semantics under with the learned style\textures

Visualizing CNN features: Look at filters

Visualization of patterns learned by the layer conv6 (top) and layer conv9 (bottom) of the network trained on ImageNet. Each row corresponds to one filter.

The visualization using "guided backpropagation" is based on the top 10 image patches activating this filter taken from the ImageNet dataset.

Dosovitskiy et al., "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Feature Inversion

Reconstruct "natural image" from given features $\phi_0$

$$x^* = \arg\min_{x \in \mathbb{R}^{B \times W \times C}} \ell(\Phi(x), \Phi_0)$$

$\ell(\Phi(x), \Phi_0) = ||\Phi(x) - \Phi_0||^2$


Feature Inversion

Feature Inversion

Feature Inversion
Feature Inversion

Reconstruct "natural image" from given features $\Phi_0$

$$x^* = \arg \min_{x \in \mathbb{R}^{H \times W \times C}} \ell(\Phi(x), \Phi_0) + \lambda \mathcal{R}(x)$$

Total Variation regularizer (encourages spatial smoothness)

$$\mathcal{R}_{TV}(x) = \sum_{i,j} \left( (x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)^{\frac{1}{2}}$$


Feature Inversion

Reconstructions from the representation after last pooling layer (immediately before the first Fully Connected layer)


Feature Inversion

Higher layers code semantic content

Higher layers are less sensitive to changes in color, texture, and shape


Feature Inversion

Captures content

Stylianou et al, "Image Style Transfer using Convolutional Neural Networks", CVPR 2016
CNN Transfer learning motivation

- Capture semantics – high layers
  - Detailed pixel information is lost

How to capture style?

$$\sum = \left( \langle f_2, f_1 \rangle \right)$$

Feature correlation matrix – Gram matrix
Spatial data is lost
Neural Style Transfer

1. Pretrain CNN
2. Compute features for content image
3. Compute Gram matrices for style image
4. Randomly initialize new image
5. Forward new image through CNN
6. Compute style loss (L2 distance between Gram matrices) and content loss (L2 distance between features)
7. Loss is weighted sum of style and content losses
8. Backprop to image
9. Take a gradient step
10. GOTO 5
Choosing the content features layer

Same parameters choice: $\alpha = 1 \times 10^{-3}$

Conv2_2

Neural Style Transfer results

$\text{Jesuiten III by Lyonel Feininger, 1915.}$

$\text{The Shipwreck of the Minotaur by J.M.W. Turner, 1805.}$
Neural Style Transfer results

The Starry Night by Vincent van Gogh, 1889. Der Schrei by Edvard Munch, 1893

Fast Style Transfer

Problem: Style transfer is slow; need hundreds of forward + backward passes of VGG

Solution: Train a feedforward network to perform style transfer!

Fast Style Transfer

- Train a feedforward for each style
- After training, stylize images using a single forward pass

Fast Style Transfer - Training

Pretrained VGG

Trained network

Fast Style Transfer

Style
The Muse,
Pablo Picasso,
1935

Gatys
Johnson

Fast Style Transfer

Style
The Simpsons

Gatys
Johnson

Fast Style Transfer

Works real-time at test-time


https://github.com/jcjohnson/fast-neural-style
Fast Style Transfer
• Train a feedforward network for each style
• After training, stylize images using a single forward pass

Fast Style Transfer of Arbitrary Style

Limited to trained styles


Fast Style Transfer of Arbitrary Style
• The Encoder
• Style transform
• The Decoder

Single Layer activations (relu3_1)


Fast Style Transfer of Arbitrary Style

Encoder
Style transfer
Decoder

The Encoder
The Decoder

Optimization

VGG19
Mix
Inverse

CNN Decoder

Fast Style Transfer of Arbitrary Style – Style swap


Fast Style Transfer of Arbitrary Style – NN swap


Where is the Gram matrix?

Hand crafter VGG activations

Content image
Style image
VGG activation function
Natural image prior

Fast Style Transfer of Arbitrary Style


**Runtime:**

<table>
<thead>
<tr>
<th>Method</th>
<th>N. Iters.</th>
<th>Time/Iter. (s)</th>
<th>Total (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gatys et al. [111]</td>
<td>500</td>
<td>0.0054</td>
<td>50.20</td>
</tr>
<tr>
<td>Style Swap (Ours)</td>
<td>100</td>
<td>0.0666</td>
<td>4.66</td>
</tr>
<tr>
<td>Style Swap (ImNet)</td>
<td>1</td>
<td>1.2483</td>
<td>1.25</td>
</tr>
</tbody>
</table>

Tune patch size to tradeoff between content structure and style texture.


**Style**

*La Muse*, Pablo Picasso, 1935

Gatys *et al.*
Fast Style Transfer of Arbitrary Style

Gatys et al., "Image Style Transfer using Convolutional Neural Networks", CVPR, 2016

Photo Realism?

Gatys et al., "Image Style Transfer using Convolutional Neural Networks", CVPR, 2016

Thanks!