Image Style Transfer
Older Methods of Texture Generation

A. Efros, W. Freeman, Image Quilting for Texture Synthesis and Transfer, 2001

M. Ashikhmin, Synthesizing Natural Textures, 2001
A VGG Network Was Used to Obtain Results

https://www.cs.toronto.edu/~frossard/post/vgg16/
Image Style Transfer Algorithm

\[ E_L = \sum (G^L - A^L)^2 \quad \text{and} \quad \mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style} \]

\[ \frac{\partial E_L}{\partial F^L} \quad \frac{\partial E_L}{\partial F^{L-1}} \]

\[ \mathcal{L}_{content} = \sum (F^l - P^l)^2 \]

\[ \mathcal{L}_{style} = \sum w_l E_l \]

\[ \vec{a} = \Rightarrow \vec{x} = \Rightarrow \vec{p} = \]

\[ \vec{x} := \vec{x} - \lambda \frac{\partial \mathcal{L}_{total}}{\partial \vec{x}} \]
doi: 10.1109/CVPR.2015.7299155
Higher Levels of the Network Preserve Semantic Content
Content Representation

To visualize the image information at each layer, a gradient descent is performed on a white noise image to find another image that matches the feature response of the original.

Loss Function for Image Content

\[
L_{\text{content}}(\bar{p}, H, l) = \frac{1}{2} \sum_{i,j} (F_{i,j}^l - P_{i,j}^l)^2
\]

Derivative with Respect to Activations

\[
\frac{\partial L_{\text{content}}}{\partial F_{i,j}^l} = \begin{cases} 
(F_l^l - P_l^l)_{i,j} & \text{if } F_{i,j}^l > 0 \\
0 & \text{if } F_{i,j}^l < 0
\end{cases}
\]
Style Representation
Style Representation

- Compute activations @ desired layers
- Calculate Gram Matrices of activations

\[
G(v_1, \ldots, v_k) = \begin{pmatrix}
    \langle v_1, v_1 \rangle & \cdots & \langle v_1, v_k \rangle \\
    \vdots & \ddots & \vdots \\
    \langle v_k, v_1 \rangle & \cdots & \langle v_k, v_k \rangle
\end{pmatrix}.
\]

- Style Loss function:

\[
E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2
\]

- Total Style Loss:

\[
\mathcal{L}_{\text{style}}(\bar{a}, \bar{x}) = \sum_{l=0}^{L} w_l E_l
\]
Training the Network

- Total Loss:

\[ \mathcal{L}_{\text{total}}(\tilde{p}, \tilde{a}, \tilde{x}) = \alpha \mathcal{L}_{\text{content}}(\tilde{p}, \tilde{x}) + \beta \mathcal{L}_{\text{style}}(\tilde{a}, \tilde{x}) \]

- L-BFGS Optimization

https://www.youtube.com/watch?v=1A6l0vcMNdc
Why L-BFGS?

Content image

Style image

Gradient Descent

Adadelta

Adagrad

Adam

RMSprop

L-BFGS

Loss vs Wall time (seconds)

https://blog.slavv.com/picking-an-optimizer-for-style-transfer-86e7b8c8ba84b
Example

Iteration: 0
Weight Adj. for Content: 5.18e-11, Style: 2.14e-29, Denoise: 5.61e-86

Content  Mixed  Style

Iteration: 10
Weight Adj. for Content: 2.79e-11, Style: 4.13e-28, Denoise: 1.25e-97

Content  Mixed  Style

Iteration: 50
Weight Adj. for Content: 2.75e-11, Style: 1.12e-27, Denoise: 1.24e-97

Content  Mixed  Style

Final image:

Style Matching vs. Content

\[ L_{\text{total}}(\tilde{p}, \tilde{a}, \tilde{x}) = \alpha L_{\text{content}}(\tilde{p}, \tilde{x}) + \beta L_{\text{style}}(\tilde{a}, \tilde{x}) \]
Effect of Different Layers for Content

- Lower layers vs. Higher Layers
- Fine Structure vs. Style Blending

\[(\alpha/\beta = 1 \times 10^{-3})\]
Discussion

Strengths
- New process of image generation combining features of 2 images from different layers of a CNN
- “I don’t care what it was designed to do, I care about what it can do”

Weaknesses
- System is not perfect, photorealism is still a hard area to handle
- Can be used to create fake content
- Resolution is limited (better in more recent examples)
Photorealistic Image Style Transfers

Original photo  Reference photo  Result

Future Research

- Removing noise to achieve photorealistic image transfers
- Artistic outlets for creativity
Thank You