Neural Style Transfer: Implementation, Result and its Applications

Qingying Luo

Duke
PRATT SCHOOL of ENGINEERING

Overview

In this experiment, we tried to transfer the style of one image into content of another image to create an artificial new one, following the procedures described by Gatys in his paper. We started from a pretrained network VGG19, which had 16 convolutional layers and learned in different scales, to help us separate image content and image style. We input the content and style images and get the activations as targets at certain layers. We left all weights in pretrained model unchanged, and only backpropagated on the input pixels. Input can start from white noise, content, or style image. The resulted image can be modified by controlling the level of content and style in the loss function.

- Content extraction focused on higher layer who preserved high-level feature information for actual content of the image instead of exact pixel values nearby.
- Style extraction used correlation between different filter responses in the same chosen layer. The dot product was able to capture the texture, like colors, and ignored the actual global arrangement.

Implementation in PyTorch

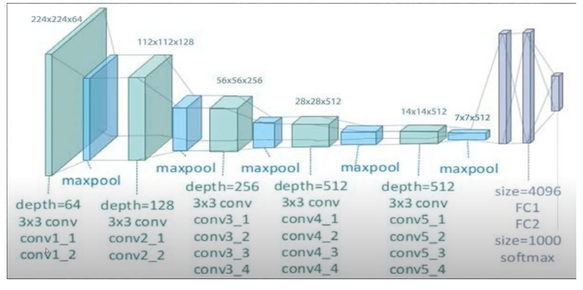
Model Setting

- Pretrained Model: VGG19
- Content Layer: conv4_2
- Style Layers: conv1_1, conv2_1, conv3_1, conv4_1, conv5_1
- Optimizer: L-BFGS
- Alpha (content weight): 1
- Beta (style weight): 8 x 1e4
- Epoch: 4 x 1e3
- Loss function:

$$\mathcal{L}_{\text{content}}(\vec{p},\vec{x},l)=\frac{1}{2}\sum_{i,j}\left(F_{ij}^{l}-P_{ij}^{l}\right)^{2}$$

$$E_l=\frac{1}{4N_l^2M_l^2}\sum_{i,j}\left(G_{ij}^{l}-A_{ij}^{l}\right)^{2}\qquad \mathcal{L}_{\text{style}}(\vec{a},\vec{x})=\sum_{l=0}^{L}w_lE_l,$$

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



Internal architecture of VGG19

Result and Implications

Different content and style pairs

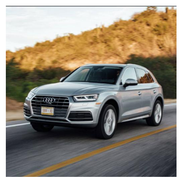


Figure 1: Car

Dock

Figure 2: French Bulldog

Picasso Painting

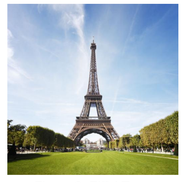


Figure 3: Eiffel Tower

Starry Night

Style layers from Figure 2 are chosen from the shallow layers, which only provided information on the color pattern and not preserved much of the detailed texture information. Compared with Figure 1 and 3, where we also extracted higher layers for style, the resulting images have reflected more feature information and textures from the style image.

Same style with different complexities in content

I used the same starry night style image on different content images with varied level of complexities.

Duke chapel with trees, grass and chair blocking in the front, making the picture more complex than the single Eiffel Tower image

A simple portrait of Lily Collin, with a pure black background, result in a low complexity

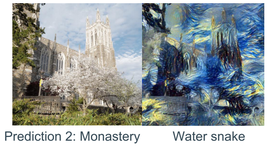
When the complexity in content image increased, say the objects in the image became crowded, like the Duke Chapel example, the neural style transfer algorithm would perform worse, since it might not be able to detect the exact boundary of different objects when it tried to blend in the style.

On the contrary, if the background was clear and the object was centered, large and obvious, then the resulting image could preserve most of the details from the content image and blended in the style naturally.

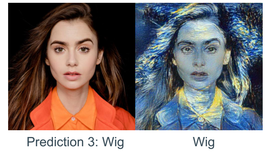
Further Applications

See how VGG19 classify the generated artificial images

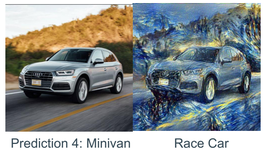
Prediction 1: Beacon Water snake



Prediction 2: Monastery Water snake



Prediction 3: Wig Wig



Prediction 4: Minivan Race Car

VGG19 also took style, color, texture into consideration when classifying. It did well on the original content images. However, when new style blended in, it tended to bias toward the style representation, especially when the original content image is not clear enough. If we used these as data augmentation, it would make the model tend to focus more on the content and partly ignored the texture information.

References

Gatys et al., “Image Style Transfer Using Convolutional Neural Networks,” (https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf))