A Neural Algorithm of Artistic Style

Anders Museth

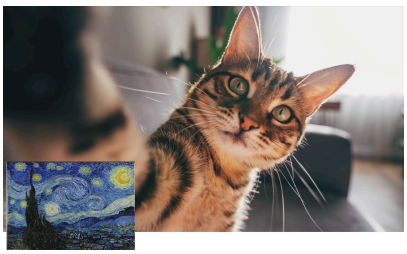
The Style Transfer algorithm is an elegant and straightforward concept. It involves utilizing a pre-trained image classification network —VGG19 in our case—that employs Convolutional Neural Networks (CNNs). These CNNs serve as filters for performing style transfer. The process begins by taking two inputs: a content image and a style image. These images are adjusted to match the standard deviation and mean of the VGG network's training data. Once adjusted, they are passed through the VGG network, during which we record the convolution responses at each layer. Next, we calculate the Gram matrices (which is a similarity metric) for the style responses at the selected layers. Following this, we replicate the content image and similarly process it through the VGG network, recording its convolution responses. The final step involves calculating the loss of this replicated image, using a method detailed below:

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

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

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$

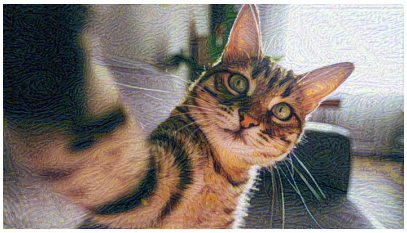
In this context, 'G' and 'A' represent the Gram matrices of the replicated image and the style image respectively. The variable 'w' serves as a hyperparameter that determines the relative importance of each layer in the computation. Additionally, 'N' denotes the number of filters, while 'M' represents the product of the height and width of the convolutional responses. 'F' denotes the feature response of the replicated image, while 'P' represents the feature response of the original content image. The next crucial step involves aggregating the two types of losses—content loss and style loss—each weighted by a specific hyperparameter. In my implementation, I have opted for a weight of 1 for content loss and 1e6 for style loss. This weighted combination of losses is a key factor in achieving a balanced style transfer. The final step is to apply gradient descent to the replicated image. This iterative optimization process gradually adjusts the image, aligning its style with that of the style image.



Style and content images

Style and content images

Style and content images



100 steps of gradient decent

100 steps of gradient decent

100 steps of gradient decent

500 steps of gradient decent

500 steps of gradient decent

500 steps of gradient decent

1000 steps of gradient decent

1000 steps of gradient decent

1000 steps of gradient decent

3000 steps of gradient decent

3000 steps of gradient decent

3000 steps of gradient decent