Neural Style Transfer

Junyang Luo

002103873

# Style Transfer Model Architecture

# Introduction

Neural style transfer (NST) refers to a class of software algorithms that manipulate digital images, or videos, in order to adopt the appearance or visual style of another image. NST algorithms are characterized by their use of deep neural networks for the sake of image transformation. Common uses for NST are the creation of artificial artwork from photographs, for example by transferring the appearance of famous paintings to user-supplied photographs. Several notable mobile apps use NST techniques for this purpose, including Deep-Art and Prisma. This method has been used by artists and designers around the globe to develop new artwork based on existent style(s).



## Earlier style transfer algorithms

NST is an example of image stylization, a problem studied for over two decades within the field of non-photorealistic rendering. The first two example-based style transfer algorithms were image analogies and image quilting. Both methods were based on patch-based texture synthesis algorithms.

Given a training pair of images–a photo and an artwork depicting that photo–a transformation could be learned and then applied to create new artwork from a new photo, by analogy. If no training photo was available, it would need to be produced by processing the input artwork; image quilting did not require this processing step, though it was demonstrated on only one style.

## NST

NST was first published in the paper "A Neural Algorithm of Artistic Style" by Leon Gatys et al., originally released to ArXiv 2015, and subsequently accepted by the peer-reviewed CVPR conference in 2016. The original paper used a VGG-19 architecture that has been pre-trained to perform object recognition using the ImageNet dataset.

In 2017, Google AI introduced a method that allows a single deep convolutional style transfer network to learn multiple styles at the same time. This algorithm permits style interpolation in real-time, even when done on video media.

## Formulation

The process of NST assumes an input image and an example style image .

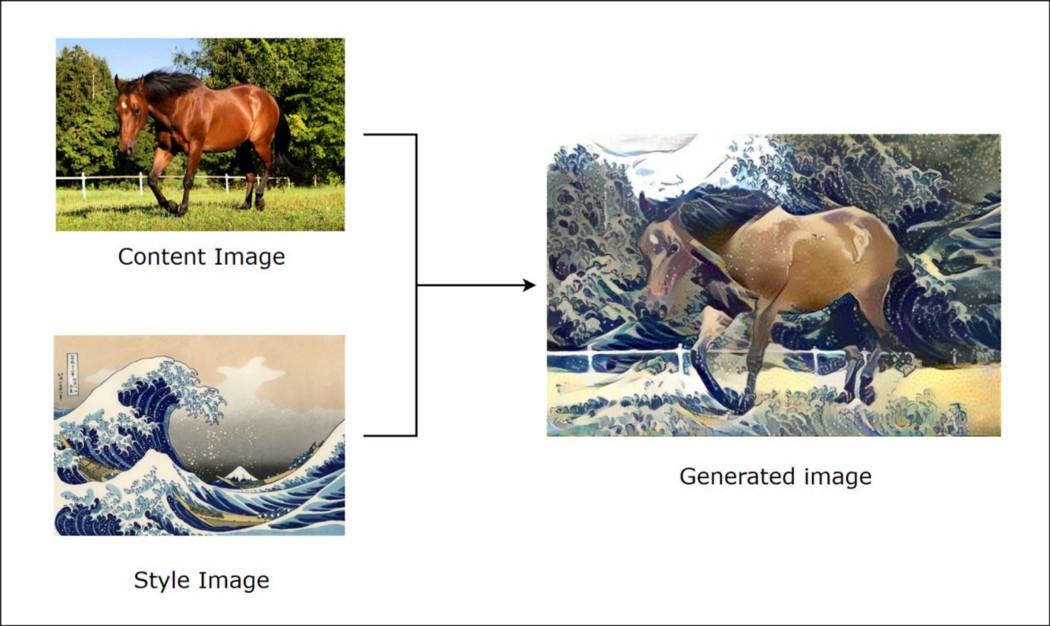
The image is fed through the CNN, and network activations are sampled at a late convolution layer of the VGG-19 architecture. Let be the resulting output sample, called the 'content' of the input .

The style image is then fed through the same CNN, and network activations are sampled at the early to middle layers of the CNN. These activations are encoded into a Gramian matrix representation, call it to denote the 'style' of .

The goal of NST is to synthesize an output image that exhibits the content of applied with the style of , i.e. and .

An iterative optimization (usually gradient descent) then gradually updates to minimize the loss function error:

where is the L2 distance. The constant controls the level of the stylization effect.



## Training

Image is initially approximated by adding a small amount of white noise to input image and feeding it through the CNN. Then we successively backpropagate this loss through the network with the CNN weights fixed in order to update the pixels of . After several thousand epochs of training, an (hopefully) emerges that matches the style of and the content of .

Algorithms are typically implemented for GPUs, so that training takes a few minutes.

# Method

- [NumPy](<https://numpy.org/>)

- [TensorFlow](<https://www.tensorflow.org/>)

- [Keras](<https://keras.io/>)

- [OpenCV](<https://opencv.org/>)

# Results

Image (input image):



Image (style image):



Image (output image):



# References

[1] <https://en.wikipedia.org/wiki/Neural_style_transfer>

[2] <https://www.youtube.com/watch?v=R39tWYYKNcI>

[3] <https://www.youtube.com/watch?v=bFeltWvzZpQ>

[4] <https://www.youtube.com/watch?v=imX4kSKDY7s>

[5] <https://github.com/jcjohnson/fast-neural-style>

[6] <https://github.com/titu1994/Neural-Style-Transfer>