A Neural Algorithm of Artistic Style

Liyuan Su   Elaheh YousefiAmiri

February 1, 2019

Overview

Problem Statement

The generated image B combines the "content" of the image A with the "style" of image S.

Landscape (content) + Scream (style)

- Traditional Methods: Non-parametric
- Deep Learning based Methods
  - Optimization
  - Convolutional Neural Networks(Gatys et al.)
  - Feed-forward(Johnson et al.)

Methods of Artistic Styles

Optimization Method

Optimization

"Neural style transfer used an optimization technique that is, starting off with a random noise image and making it more and more desirable with every training iteration of the neural network."

Picasso Dancer

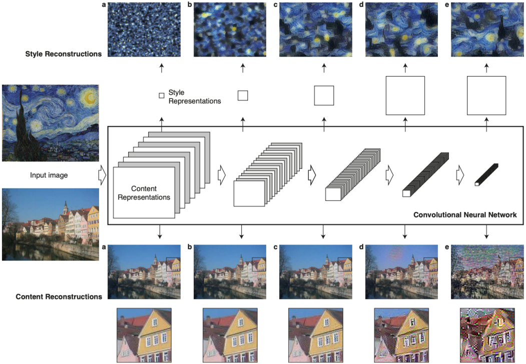
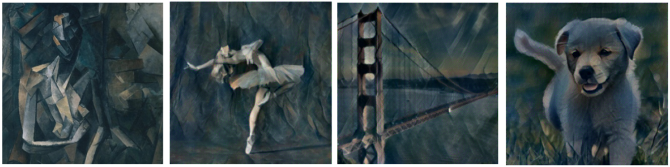


Figure: Style and content representations taken at each layer of a Neural Network.Image from(Gatys et)

Feed-Forward

"By pre-training a feed-forward network rather than directly optimizing the loss functions."



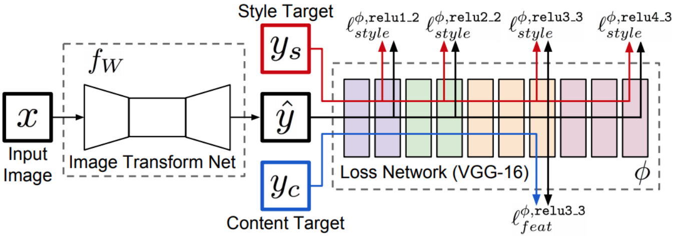


Figure: "System overview. We train an image transformation network to transform input images into output images. We use a loss network pretrained for image classification to define perceptual loss functions that measure perceptual differences in content and style between images. The loss network remains fixed during the training process." Figure from Johnson et al

Deep image representations

- Content representation
- Style representation
- Style transfer

Define the squared-error loss between the two feature representations

$$L_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} (F^l_{ij} - P^l_{ij})^2$$

The derivative of this loss with respect to the activations in layer l equal

$$\frac{\partial L_{content}}{\partial F^l_{ij}} = \begin{cases} (F^l - P^l)_{ij} & F^l_{ij} > 0, \\ 0 & otherwise \end{cases}$$

from which the gradient with respect to the image $\vec{x}$ can be computed using standard error back-propagation.

Feature correlations are given by the Gram matrix $G^l \in R^{N_l \times N_l}$,where G^l_{ij} is the inner product between the vectorised feature maps i and j in layer l:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}.$$

The contribution of layer l to the total loss is then:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G^l_{ij} - A^l_{ij})^2$$

and the total style loss is

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

where w_l are weighting factors of the contribution of each layer to the total loss

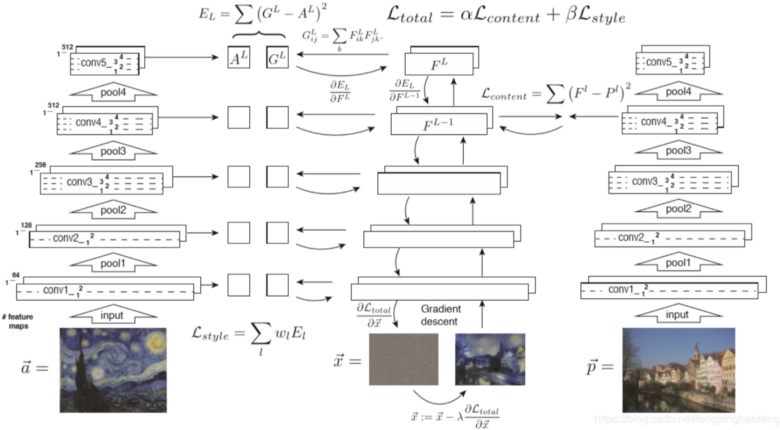
The derivative of E_l with respect to the activations in layer l can be computed analytically:

$$\frac{\partial E_l}{\partial F^l_{ij}} = \begin{cases} \frac{1}{N_l^2 M_l^2}((F^l)^T(G^l - A^l))_{ji} & if \quad F^l_{ij} > 0, \\ 0 & otherwise \end{cases}$$

The gradients of E_l with respect to the pixel values $\vec{x}$ can be readily computed using standard error back-propagation.

The loss function we minimise is

$$L_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha L_{content}(\vec{p}, \vec{x}) + \beta L_{style}(\vec{a}, \vec{x})$$



online Aplications

Style transfer Apps:

http://deepart.io

http://www.pikazoapp.com

https://artisto.my.com (Video and Photo Editor)

References

Leon A. Gatys, Alexander S. Ecker, Matthias Bethge(2015)
A Neural Algorithm of Artistic Style
https://arxiv.org/abs/1508.06576 .

Justin Johnson, Alexandre Alahi, Li Fei-Fei(2016)
Perceptual Losses for Real-Time Style Transfer and Super-Resolution
https://arxiv.org/pdf/1603.08155.pdf

Thank You
The End