A Neural Algorithm of Artistic Style

Leon A. Gatys,Alexander S. Ecker, Matthias Bethge
University of Tubingen, Germany
Published 9/2/2015

The idea

Separating **content** from **style**

Loss function: **content**

L_{content} captures the root mean squared error between the activations produced by the generated image and the content image. But why does minimising the difference between the activations of higher layers ensure the content of the content image is preserved?

$$L_{content} = \frac{1}{2} \sum_{i,j} (A_{ij}^{l}(g) - A_{ij}^{l}(c))^2$$

There is evidence that suggests that different feature maps in higher layers are activated in the presence of different objects. So if two images have the same “content”, they should have similar activations in the higher layers.

Loss function: **style**

Style information is measured as the amount of **correlation** present between features maps in a given layer. Loss is defined as the difference of correlation present between the feature maps computed by the generated image and the style image. Mathematically, the style loss is defined as,

$$L_{style} = \sum_l w^l L^l_{style} \text{ where,}$$

$$L^l_{style} = \frac{1}{M^l} \sum_{ij} (G^l_{ij}(s) - G^l_{ij} g)^2 \text{ where,}$$

$$G^l_{ij}(I) = \sum_k A^l_{ik}(I) A^l_{jk}(I).$$

w^l (chosen uniform in this tutorial) is a weight given to each layer during loss computation and M^l is an hyperparameter that depends on the size of the l th layer

So why does this loss function define ***style***?

- The Gram matrix computed in this technique contains dot products of the feature maps at a layer. This is a correlation operation. The entries basically encode activations that co-occur.

Sidebar: **The Gram Matrix**

Given a set V of m vectors (points in $\mathbb{R}^n$), the Gram matrix G is the matrix of all possible inner products of V, i.e.,

$$g_{ij} = \mathbf{v}_i^{\mathrm{T}} \mathbf{v}_j.$$

where A^{T} denotes the transpose.

[from Wolfram MathWorld]

Final loss

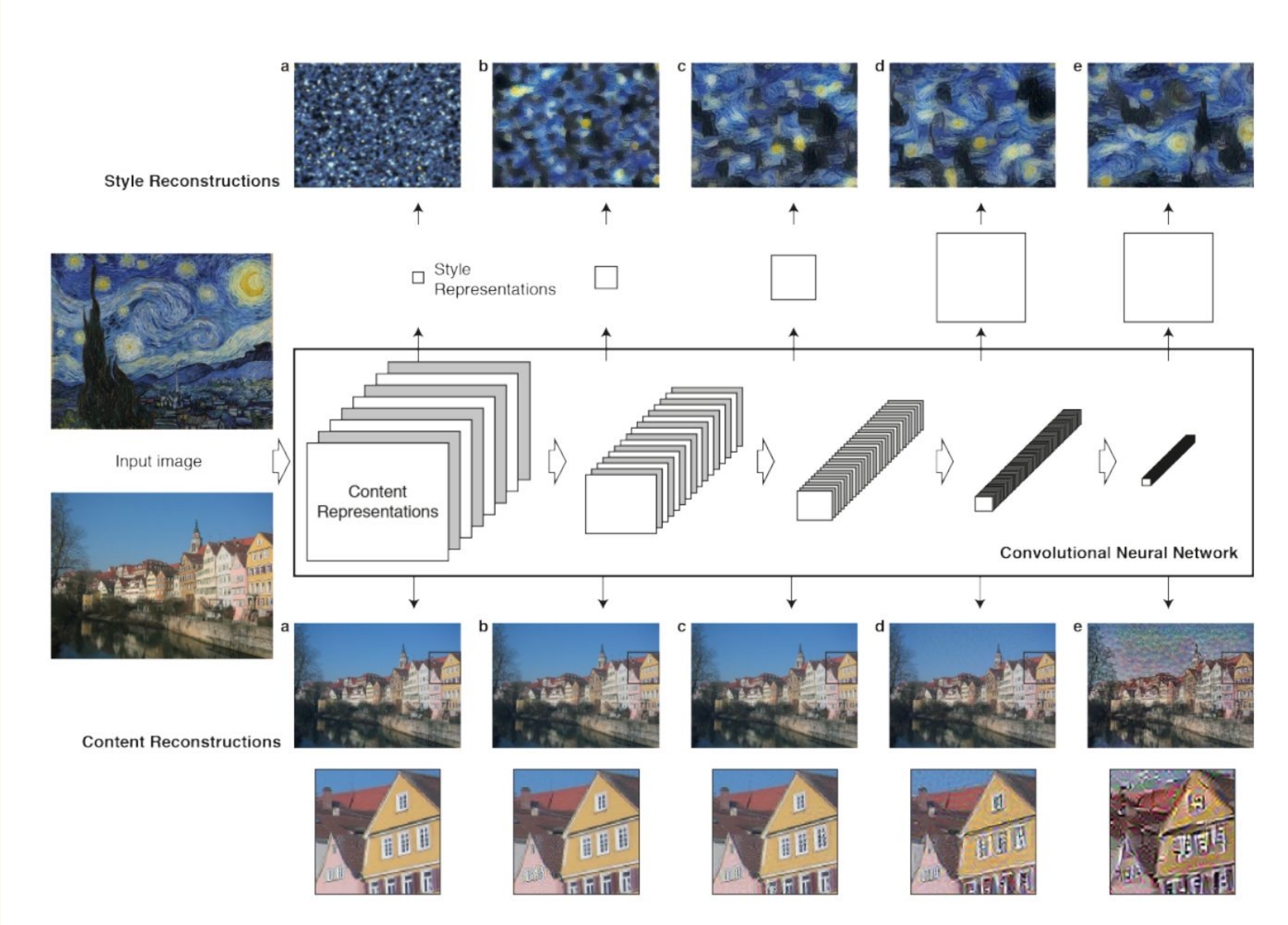
$$L = \alpha L_{content} + \beta L_{style} ,$$

α and β are tunable hyperparameters

Experiment 1: visualizing style and content independently

Do gradient descent on an image of random noise

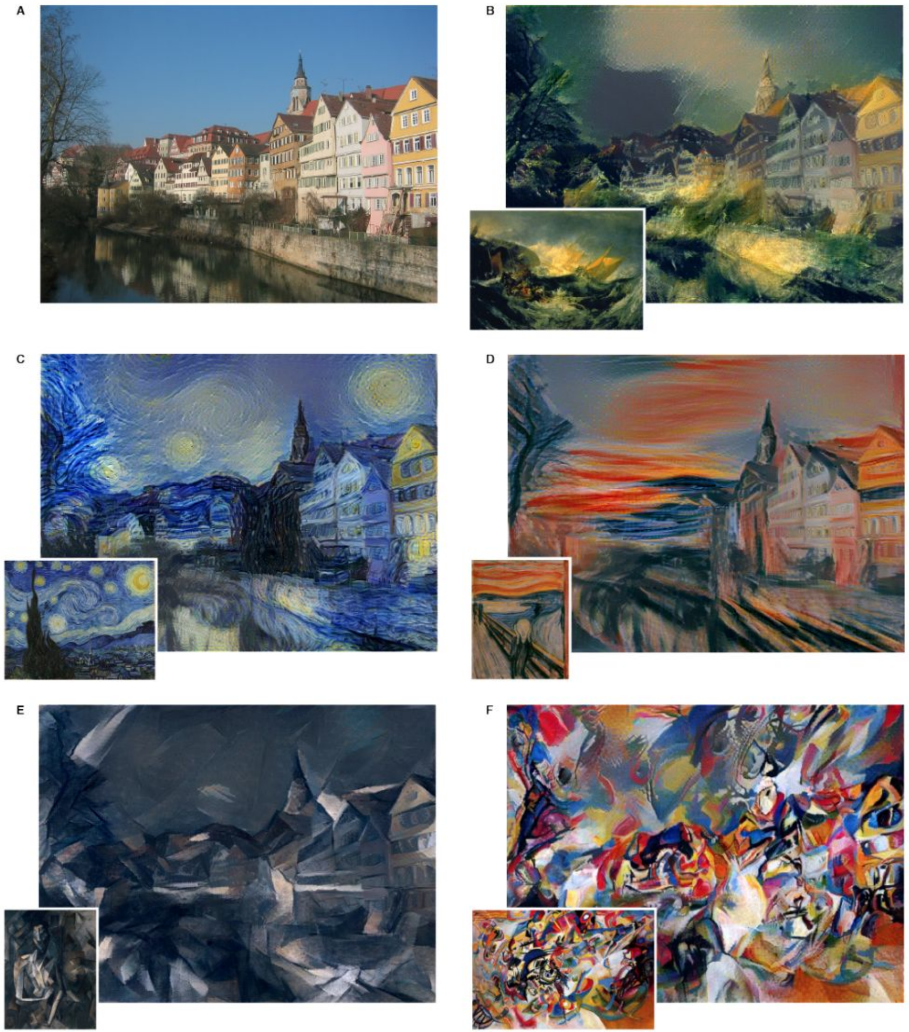
Results


Style Reconstructions
a
b
c
d
e
Style Representations
Input image
Content Representations
Convolutional Neural Network
Content Reconstructions
a
b
c
d
e

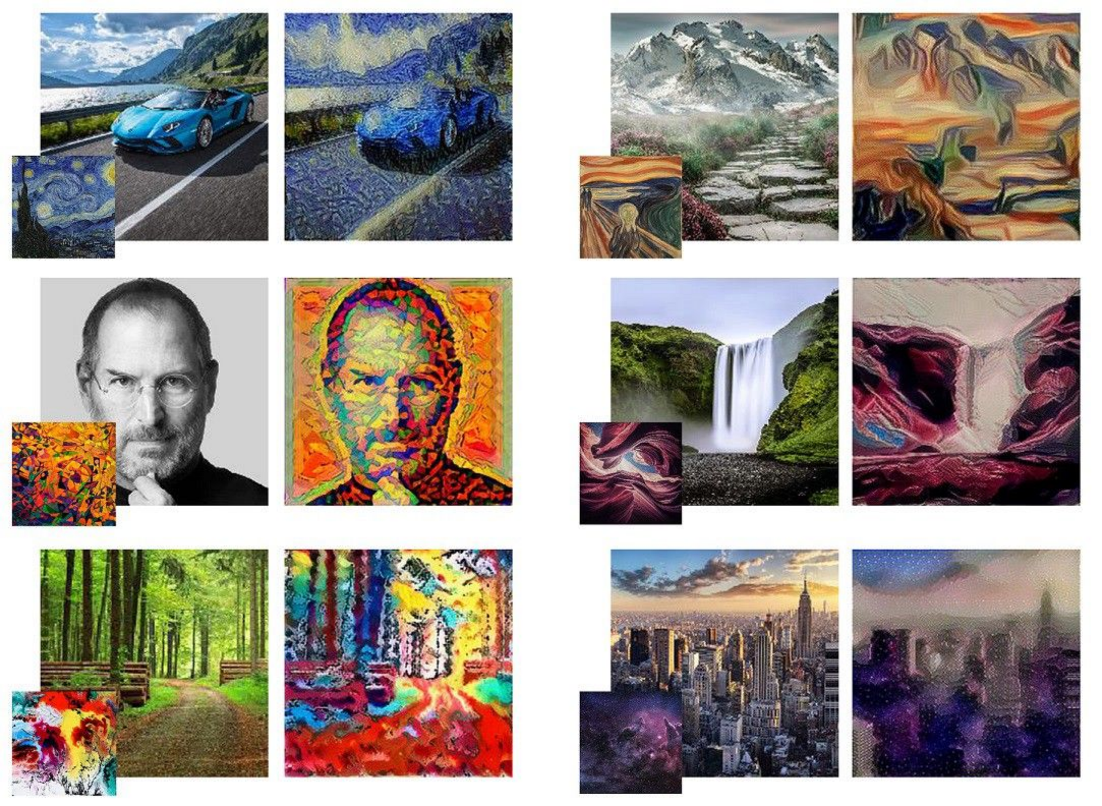
Experiment 2: combining the content of one image with the style of another

Do gradient descent on an image of random noise with the full loss function

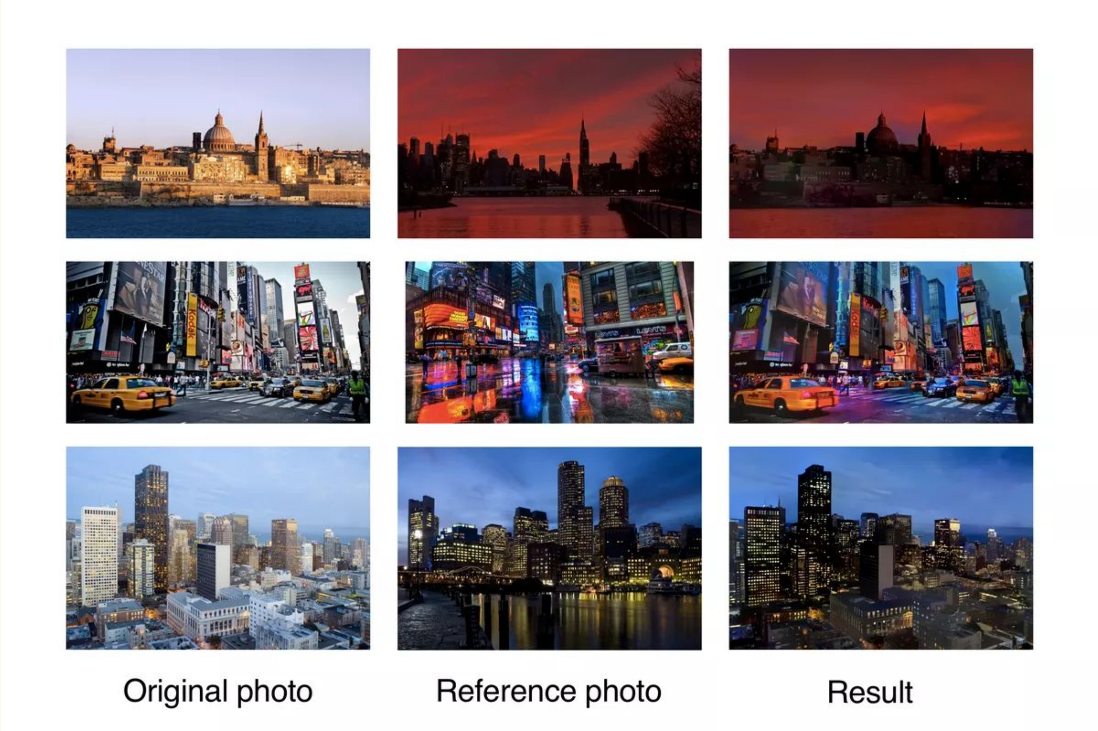
Results



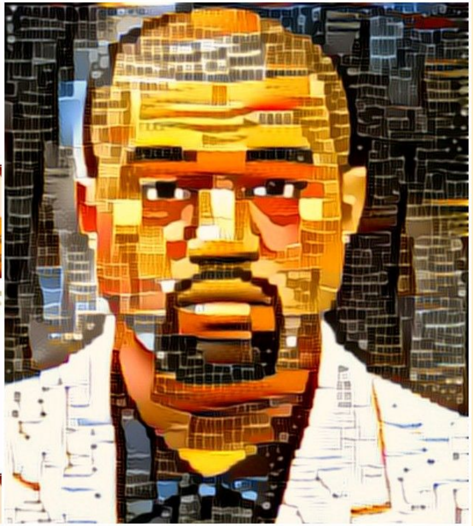
More Results

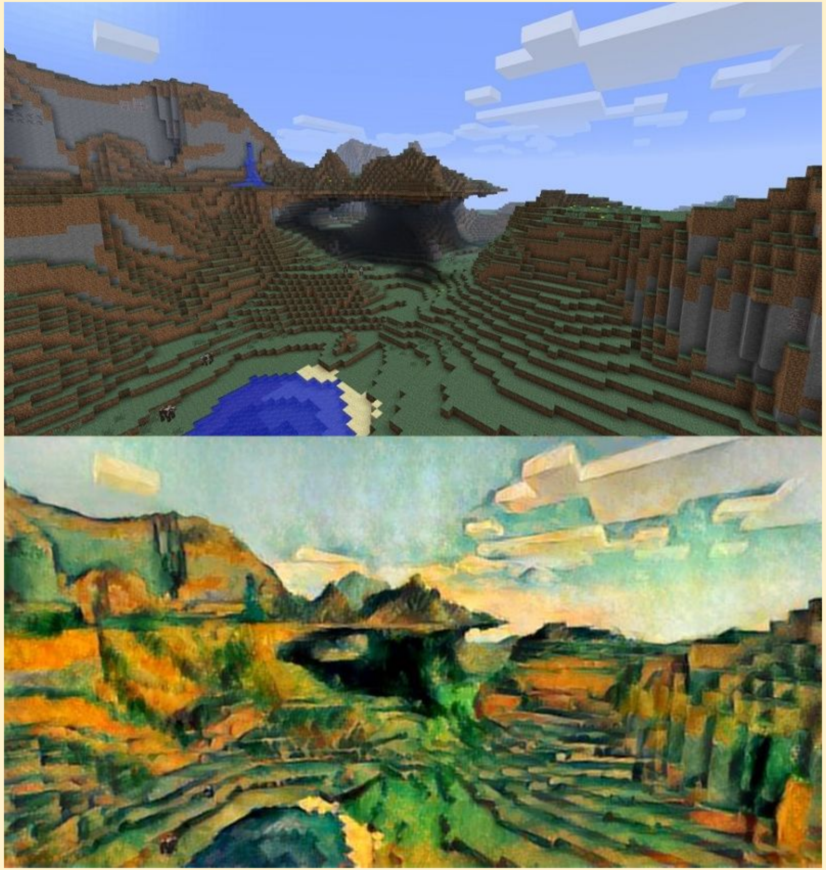


Bonus pics



Bonus bonus pics





Try our own style transfer online

https://deepart.io/