Texture Modelling Using Convolutional Neural Networks

Leon A. Gatys, Alexander S. Ecker and Matthias Bethge
University of Tübingen

1 Abstract

We introduce a new model of natural textures based on the feature spaces of convolutional neural networks optimised for object recognition. Samples from the model are of high perceptual quality demonstrating the generative power of neural networks trained in a purely discriminative fashion. Within the model, textures are represented by the correlations between feature maps in several layers of the network. We show that across layers the texture representations increasingly capture the statistical properties of natural images while making object information more and more explicit. Extending this framework to texture transfer, we introduce A Neural Algorithm of Artistic Style that can separate and recombine the image content and style of natural images. The algorithm allows us to produce new artistic imagery that combines the content of an arbitrary photograph with the appearance of numerous well-known artworks, thus offering a path towards an algorithmic understanding of how humans create and perceive artistic imagery.

2 Model

To characterise a given vectorised texture $\vec{x}$ in our model, we first pass $\vec{x}$ through the convolutional neural network and compute the activations for each layer l in the network. Since each layer in the network can be understood as a non-linear filter bank, its activations in response to an image form a set of filtered images (so-called *feature maps*). A layer with N_l distinct filters has N_l feature maps each of size M_l when vectorised. These feature maps can be stored in a matrix $F^l \in \mathcal{R}^{N_l \times M_l}$, where F^l_{jk} is the activation of the j^{th} filter at position k in layer l. Textures are per definition stationary, so a texture model needs to be agnostic to spatial information. A summary statistic that discards the spatial information in the feature maps is given by the correlations between the responses of different features. These feature correlations are, up to a constant of proportionality, given by the Gram matrix $G^l \in \mathcal{R}^{N_l \times N_l}$, where G^l_{ij} is the inner product between feature map i and j in layer l:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}. \tag{1}$$

A set of Gram matrices $\{G^1, G^2, ..., G^L\}$ from some layers $1, \ldots, L$ in the network in response to a given texture provides a stationary description of the texture, which fully specifies a texture in our model. To generate a new texture on the basis of a given image, we use gradient descent from a white noise image to find another image that matches the Gram-matrix representation of the original image. This optimisation is done by minimising the mean-squared distance between the entries of the Gram matrix of the original image and the Gram matrix of the image being generated.

Let $\vec{x}$ and $\hat{\vec{x}}$ be the original image and the image that is generated, and G^l and $\hat{G}^l$ their respective Gram-matrix representations in layer l (Eq. 1). The contribution of layer l to the total loss is then

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

and the total loss is

$$\mathcal{L}(\vec{x}, \hat{\vec{x}}) = \sum_{l=0}^{L} w_l E_l \tag{3}$$

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 and thus the gradient of $\mathcal{L}(\vec{x}, \hat{\vec{x}})$, with respect to the pixels $\hat{\vec{x}}$ can be readily computed using standard error back-propagation.