Neural-Style-Transfer

Welcome to the fun tutorial that aims for an elaborate discussion of Keras Implementation of Neural Style Transfer from the paper: A Neural Algorithm of Artistic Style

Goal: Creation of Artistic images which are aesthetically pleasing to our eyes ;)

Use: Neural Representations (from *Convolutional Neural Networks*)

Task: separate and recombine *content* & *style* of arbitrary images

Let us check the example below:

We notice that it somehow extractes low level features like the colour and texture from an artistic image `starry_night.jpg` (that we'll call the style image, `s`) and apply it to a more semantic, higher level features like a `barking dog's face` on first image (that we'll call the content image, `c`) and arrive at the style-transferred image, `x`.
Both style and content statistics are obtained from a deep convolutional network (CNN) pre-trained for image classification.

Convolutional Neural Networks consist of *layers* of small computational units that process visual information hierarchically in a feed-forward manner.

Each layer in CNN has filters that extracts certain features of the input image. The output of a given layer consists of *feature maps: differently filtered versions of the input image*

Now the question is how do we obtain statistics of content image and style image?

Content Representation When CNNs are trained on object recognition/image classification, they develop the representation of the image that makes object information greatly explicit along the processing hierarchy. Hence at higher layers the representations care highly about the actual *content* of the image like *objects and their arrangements* compared to their detailed pixel values. We can reconstruct the image/spatial information from the feature maps/feature responses in higher/deeper layers of the network which is usually refered to as content representation.

Style Representation We need to capture *texture* information of the *style* image. For this we build *feature space* on top of the filter responses in each layer of the network. Feature space consists of correlations between the different filter responses, where the expectation is taken over the spatial extend of the input image. By including feature

correlations from multiple layers of the neural network we gain a multi-scale representation the captures the generalized appearance in terms of color and localised structures without focusing on the global arragement.

How to compute feature correlations?

Compute Gram matrix:

$G^l = \sum_k F^l_{ik} F^l_{jk}$ It is the inner product between the vectorised feature maps i and j in layer l.

Style Transfer Optimization problem:

All we need to do is to find an image $\mathbf{x}$ that differs as little as possible in terms of content from the content image $\mathbf{c}$, while simultaneously differing as little as possible in terms of style from the style image $\mathbf{s}$. In other words, we'd like to simultaneously minimise both the style and content losses.

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \left(\alpha \mathcal{L}_{\text{content}}(\mathbf{c}, \mathbf{x}) + \beta \mathcal{L}_{\text{style}}(\mathbf{s}, \mathbf{x}) \right)$$

- α: content weight
- β: style weight

Here, α and β are simply numbers that allow us to control how much we want to emphasise the content relative to the style.

Keynotes:

- Crux of the paper lies in the idea that the representations of content and style in CNN are separable. Implying that we can manipulate both representaions independently to produce new, perceptually meaningful images.
- Results are generated on the basis of the VGG-Network. We will use the feature space provided by 16 convolutional and 5 pooling layers of the 19 layer VGG-Network. Since our aim is not classification, we do not use any of the fully connected layers.

Details of Algorithm

Style transfer consists in generating an image with the same "content" as a base image, but with the "style" of a different picture (typically artistic). This is achieved through the optimization of a loss function that has 3 components: "style loss", "content loss", and "total variation loss":

- The content loss is a L2 distance between the features of the base image (extracted from a deep layer) and the features of the combination image, keeping the generated image close enough to the original one.

$$\mathcal{L}_{\text{content}}(\mathbf{c}, \mathbf{x}, \mathbf{l}) = \frac{1}{2} \sum_{i,j} (F^l_{ij} - P^l_{ij})^2$$

where P^l and F^l are feature representation at layer l for content image $\mathbf{c}$ and generated image $\mathbf{x}$ respectively

- The style loss consists of a sum of L2 distances between the Gram matrices of the representations of the style image and the image to be generated, extracted from different layers of a convnet (trained on ImageNet). The general idea

is to capture color/texture information at different spatial scales (fairly large scales --defined by the depth of the layer considered).

The contribution of layer l to the style loss is,

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

where A^l and G^l are respective style representations of the original style image and the image to be generated.

- N_l = number of channels at layer l
- M_l = size of the image (height * width) at layer l

and total loss is

$$\mathcal{L}_{\text{style}}(\mathbf{s}, \mathbf{x}) = \sum_{l=0}^{L} w_l E_l$$

where w_l are weighing factors of the contribution of each layer to the total loss which is given by $\frac{1}{num-of-active-layers-with-a-nonzero-lossweight-w_l}$

- The total variation loss imposes local spatial continuity between the pixels of the combination image, giving it a visual coherence.

We are making improvements to our algorithm (proposed by Gatys et al. (2015) as described above) as suggested by the paper:

Improving the Neural Algorithm of Artistic Style

Improvement 3.1 in paper : **Geometric Layer weight adjustment for Style inference (Improvement 4 in the code)**
Here we imporve the style loss by performing geometric weighted scaling for each of the layers used in style. IT indicates that the most important style properties like color, patterns, textures are captured by the bottom layers.

Improvement 3.2 in paper : **Using all layers of VGG-16 for style inference (Improvement 2 in the code)**
To enrich the style representation we calculate Gram matrices for all 16 convolution layers of VGG-16.

Improvement 3.3 in paper : **Activation Shift of gram matrix (Improvement 1 in the code)**
Using shifted activations when computing Gram matrices helps eliminate sparsity and make individual entries of the matrix more informative.

Improvement 3.5 in paper : **Correlation Chain (Improvement 3 in the code)**
Here correlation of features belonging to different layers is targetted to capture more feature interactions. Chained style representation: $G^{l,l-1}$ for $l = 2, ...16$ is followed. This way the correlation of immediated neighbors are considered

For implementation check the functions of gram_matrix() and style_loss() in the code (Notebook).

Masked Style Transfer is based on the paper:

Show, Divide and Neural: Weighted Style Transfer

Masked Style Transfer utilizes masks to determine which area is affected by the style.

White color in the mask is 255, which is normalized to 1.0, whereas black color is near 0 and normalized to 0.0. These values are multiplied by each filter in the VGG-19 network(you will see it as we go further) for style. Giving an example of the last 5-4 layer, it has 512 channels, which are all multiplied by the mask image.

Wherever the mask was white, the channel values remain the same. However wherever the mask was black, the channel values are forced to become 0 as well. Therefore the gradients at that blackened region are also zero. This indicates that the style transfer at the blacked region is 0 % as well.

A note on mask images:

- They should be binary images (only black and white)
- White represents parts of the image that you want style transfer to occur
- Black represents parts of the image that you want to preserve the content
- Be careful of the order in which mask images are presented in Multi Style Multi Mask generation. They have a 1 : 1 mapping between style images and style masks.

I have created mask images for all the tutorial examples using Photoshop

At the end of the tutorial we will see an option for color preservation which is based on the paper:

Preserving Color in Neural Artistic Style Transfer

Hope you enjoy the tutorial!