Neural Style Transfer

Marija Lakić

School report for course Computational Intelligence
Faculty of Mathematics, University of Belgrade

July 28, 2021

Contents

1 Introduction

Neural Style Transfer represents a texture transfer problem solved using Convolutional Neural Networks. Two images are provided, usually a photograph that represents a content image, and an artwork image that represents the style image.

Objective of this problem is to transfer the texture of style image to content image, creating a combined image that retains the object placement of content image. End result should be content image drawn in artistic style of style image.

In original work [1] it is mentioned that this problem offers a path forward to an algorithmic understanding of how humans create and perceive artistic imagery.

2 Solution

2.1 General idea

Process of combining the two images consists of three steps: extracting image features, computing a spatial summary statistics and gradient descent on white-noise image.

Extracting image features is achieved by passing the images through pre-trained VGG19 neural network for object recogniton on Imagenet dataset. Spatial summary statistic is computed on extracted feature responses to obtain a stationary description of the image[3]. Finally, final image is created by performing gradient descent on white-noise image.

2.2 Spatial summary statistics

2.2.1 Content loss function

To visualise the image information that is encoded at different layers of the hierarchy one can perform gradient descent on a white noise image to find another image that matches the feature responses of the original image[2].

Each layer l in CNN defines its filters. A layer with N_l filters has N_l feature maps of size M_l, where M_l is height times width of feature map. Responses in layer l can be stored in a matrix $F^l \in R^{N_l \times M_l}$, where F^l_{ij} is the activation of the filter i in position j of layer l. Let $\vec{p}$ be the original image and $\vec{x}$ the image that is generated, P_l and F_l their feature representations in layer l.

For reconstructing content representation following loss function is used:

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

2.2.2 Style loss function

Style representations is viewed as extracting texture information. This information is obtained with a summary statistics that discards the spatial information

in the feature maps, given by the correlations between responses in different feature maps [3]. Summary statistic used for this purpose is Gram matrix $G^l \in R^{N_l \times M_l}$:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \tag{2}$$

Let $\vec{a}$ and $\vec{x}$ be the original image and image to be generated, A_l and F_l their respective Gram matrices of feature maps in layer l. The contribution of layer l to the total style loss is:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G^l_{ij} - A^l_{ij}) \tag{3}$$

and the total style loss is:

$$L_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l \tag{4}$$

where w_l are weight factors defined for each layer.

2.2.3 Total loss function

Style transfer is achieved through minimising the distance of the feature representations of a white noise image from the content representation of the photograph in one layer and the style representation of the painting defined on a number of layers of the CNN[2].

The loss function to be minimized is:

$$L_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha L_{content}(\vec{p}, \vec{x}) + \beta L_{style}(\vec{a}, \vec{x}) \tag{5}$$

Hyperparameters α and β are weighting factors for content and style representation.

3 Methods

For extracting image features pretrained VGG19 CNN is used with average pooling layers. Adam optimizer is used for style and content representation. Following original work[1], layers used for extracting style features are: *block*1*_conv*1, *block*2*_conv*1, *block*3*_conv*1, *block*4*_conv*1, *block*5*_conv*1. For extracting content features layer *block*4*_conv*2 is used.

Parameters that were tested in this work are weights of style layers w_l, style and content weight α and β, number of style layers used and learning rate of the optimizer.

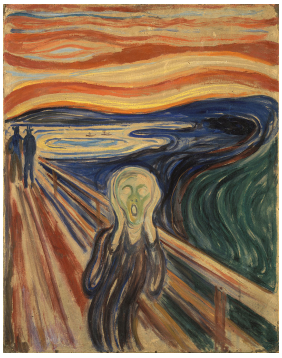
4 Results

Photograph used for representing content is Neckarfront in Tübingen. Style images used for demonstrating this solution are *The Shipwreck of the Minotaur* by J.M.W. Turner, *The Starry Night* by Vincent Van Gogh, *Der Schrei* by Edvard Munch, *Femme nue assise* by Pablo Picasso, *Composition VII* by Wassily Kandinsky.

(a) Content image, Neckarfront in Tübingen, Germany

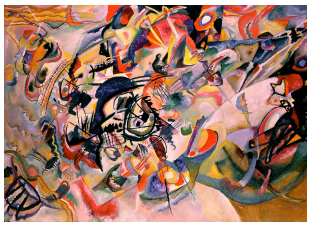
(b) *The Shipwreck of the Minotaur*, J.M.W. Turner

(c) *The Starry Night*, Vincent Van Gogh



(d) *Der Schrei*, Edvard Munch

(e) *Femme nue assise*, Pablo Picasso



(f) *Composition VII*, Wassily Kandinsky

Figure 1: Content and style images

4.1 Content representation

To generate a texture that matches the content of a given image we use gradient descent from a white noise image to find another image that matches the content representation of the original image. Figure 2 represents this process for 1000 iterations and learning rate set to 0.02.

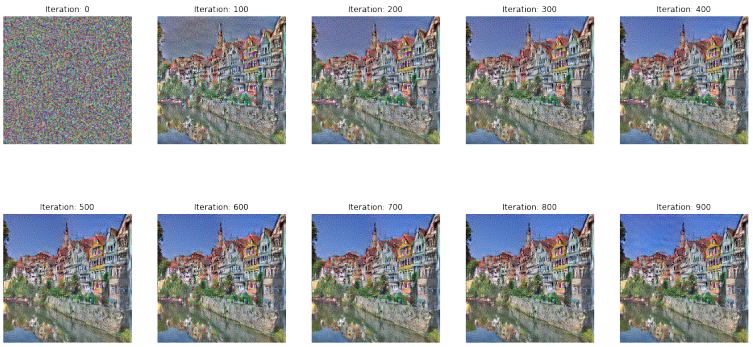


Figure 2: Content representation

4.2 Style representation

Similarly, using the same procedure over layers we use for extracting style, we get style representation.

Figure 3 shows style representation of *The starry Night*. All 5 style layers are used with equal weights ($w_l = \frac{1}{5}$). Learning rate is set to 0.02 for 1000 iterations.

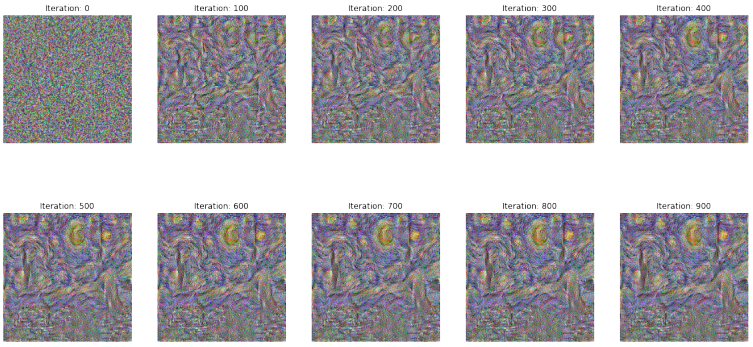


Figure 3: Style representation

Figure 4: Style representation of all style images over 500 iterations, from left to right: *The Shipwreck of Minotaur, The Starry Night, Der Schrei, Femme nue assise, Composition VII*

4.3 Style Transfer

In this section style transfer is demonstrated by varying previously mentioned hyperparameters. Figure 5 shows how different subset of style layers used influences the end result. As it was previously noticed[1], style representation is

Figure 5: Varying subset of style layers used. Number of iterations: 1000, $\frac{\alpha}{\beta} = 10^{-2}$, learning rate = 0.02

more prominent when using bigger subset of layers. Best results were noticed when using 4 or 5 of style layers.

Figure 6 shows how $\frac{\alpha}{\beta}$ ration influences the end result. The less the ratio is, the more the content of the photograph is lost. Usually, style features are more prevalent, so the best results of style transfer were achieved for $\frac{\alpha}{\beta} = 10^{-3}$ or $\frac{\alpha}{\beta} = 10^{-2}$.

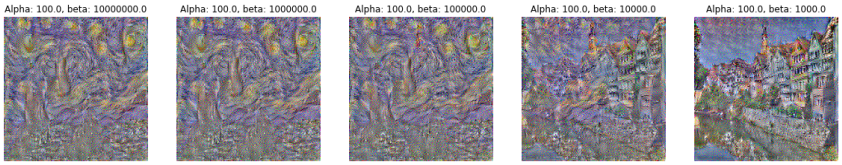


Figure 6: Varying $\frac{\alpha}{\beta}$ ratio.

Style transfer was tested for different style weights for each layer. All previously mentioned results were achieved for same style weights, where $w_l =$

$1/number\_of\_layers\_used$. Figure 7 compares results for different style weights, influencing more higher or lower layers.

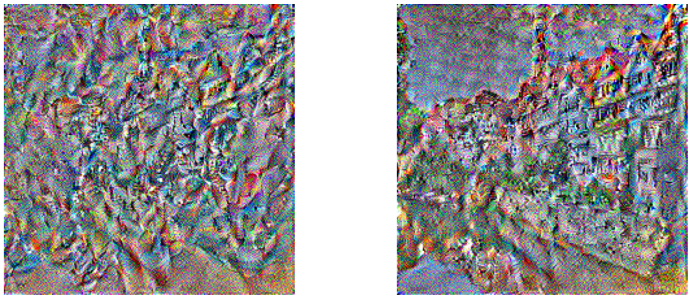


Figure 7: Varying w_l. All 5 layers were used. Image on the left was generated using array $[0.05, 0.05, 0.1, 0.8]$, while image on the right used the same array reversed. First image retains more of style features, while second image retains more of content features

For optimization Adam optimizer was used. In optimization problems, bigger learning step can potentially skip over a better solution. Here, it just brings out the style features faster, it doesn't necessarily portray style features badly.

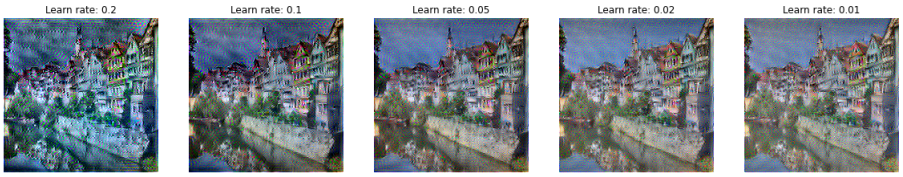


Figure 8: Varying learning rate of the optimizer

5 Conclusion

In this work, Neural Style Transfer was demonstrated following the original work[1]. However, there are many works on this subject trying to improve the solution changing the CNN parameters or summary statistic used.

Noticing how much style can be reconstructed using the layers of CNN, the only concern is adjusting the style and content factor. The biggest problem is that this is purely subjective to the observer, whether the end result should be more influenced by style or content, it can't be measured. It also depends of the artwork used. For example, reconstructing style for *Femme nue assise* by Pablo Picasso turned out to be harder than for other artworks mentioned in this report. This can be potentially improved by changing the style loss function[4].

Another improvement would be denoising the images, which wasn't the focus here.

Originally, L_BFGS was used for optimization, but Adam also gives satisfying results. Other gradient descent techniques or other optimization techniques that cover a wider search space could be tested.

References

[1] Matthias Bethge Leon A. Gatys Alexander S. Ecker. *A Neural Algorithm of Artistic Style*. 2015.

[2] Matthias Bethge Leon A. Gatys Alexander S. Ecker. *Image Style Transfer Using Convolutional Neural Networks*. 2016.

[3] Matthias Bethge Leon A. Gatys Alexander S. Ecker. *Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks*. 2015.

[4] Jiayang Liu Yanghao Li Naiyan Wang. *Demistifying Neural Style Transfer*. 2017.