Neural Style Transfer

Stefan Wezel

Abstract: *Introduced by Gatys* et al. *(2015), the field of Neural Style Transfer has not only evolved rapidly but also allowed for insights into the processes inside neural networks and into human perception. Various methods, ranging from image based to model based approaches have been introduced to alleviate the weaknesses of the original formulation. Here, we recapture the idea behind Gatys* et al. *(2015)'s algorithm and give an overview of methods, used in the current field of Neural Style Transfer.*

Contents

1 Introduction

Art has played an important role in human culture throughout most of its history [Carroll (2004)]. Despite this, little is known about what the deciding factors of what we perceive as aesthetic are. Recent progresses in Artificial Intelligence yield astonishing accuracy in computer vision tasks, leading to the impression, that Convolutional Neural Nets almost rival the perceptive prowess of the human visual cortex.
Applying the style of one image to the content of another has been a topic in the field of non-realistic rendering for more than two decades [Jing *et al.* (2019)].
Gatys *et al.* (2015)'s work showed that powerful Convolutional Neural Nets can be used to transfer arbitrary styles to any content image. Besides the visually astonishing results, their work also gives us an insight on the creation and perception of artistic images, a field where Neural Networks have not yet been as outstanding as their human counterpart.

2 Related Work

Before Neural Networks were applied to Style transfer, a popular approach to the problem existed in image-based artistic rendering [Kyprianidis *et al.* (2012)]. This area of research can be subdivided into multiple directions. **Linear Transformations**, i.e. Filters designed for image processing are used to create stylized images by Winnemöller *et al.* (2006) and Tomasi and Manduchi (1998) among others.
Another popular approach, **Stroke-Based Rendering** typically starts with a photograph on which then strokes are placed to mimic a certain style. The placement of the strokes is optimized to a given objective function some quantity that measures how similar the synthetic paining is to a specified image [Hertzmann (2003)]. Another technique is **Region-based Rendering** where an image is segmented into different regions. This enables a rendering algorithm to be sensitive to each region's specific content [Kolliopoulos (2005)]. Both these approaches do lack the ability to incorporate any arbitrary style. Therefore, Hertzmann *et al.* (2001) proposes to learn a transformation from source to target image in a supervised fashion with the **Example-Bassed Rendering** technique of Image Analogies. This, however, requires training data which may not be available.
While not having been designed for the goal of creating an artistic image, important contributions to style transfer research also came from **Texture Synthesis**. Early work there focused on pixel measurements [Julesz (1962)]. Later, filter responses played an important role in the work of Heeger and Bergen (1995) and Portilla and Simoncelli (2000). The use of summary statistics in Texture Synthesis can be noted as a precursor to the Neural Style Transfer Algorithm proposed by Gatys *et al.* (2015). There, however, not the statistics of an image, but rather the statistics of the latent representation of an image are used for measuring similarity in style.
Rather than using Descriptive Statistics, another branch of research exploits Markov Ran-

dom Fields (MRF) in a **non-parametric modeling** approach to render stylized images. A MRF assumes that each pixel is characterized solely by the pixels in its spatial neighborhood [Jing *et al.* (2019)]. Efros and Leung (1999) find pixels in the texture image by finding pixels whose neighborhoods resemble each others and then replacing them in the source image.
The topic of Neural Style Transfer can also be linked to **Image Reconstruction** where rather than encoding images into latent representation, the goal is to create an image from given information. An approach, proposed by Mahendran and Vedaldi (2015) is able to generate images by optimizing latent representations. Given random noise, the algorithm iteratively optimizes the image up to a point where its latent representation matches that of those, generated by a convolutional net. This is computationally expensive because a new training process is required for every image. In order to generate images faster, Dosovitskiy and Brox (2016) propose to train a generative model to produce an image. Then, after a training stage, images can be computed in real-time, thus shifting the computational effort.

3 Original Algorithm of Neural Style Transfer

Using summarizing statistics from the feature representation of CNNs, Gatys *et al.* (2015) propose an image based approach to Style Transfer. In their algorithm a given image is optimized to match two source images, one being responsible for the content and the other for the style of the image to optimize.

3.1 Setting

Given two images, one responsible for content, and one for style, the goal of Neural Style Transfer is to create a stylized image that matches the content images content and the style images style. The stylized image is iteratively optimized according to two loss terms.

3.2 VGG-Net

Convolutional Neural Networks have proven to represent features of input images efficiently once trained for a task like image classification. An architecture that became popular after winning the ILSVRC localization task in 2014 is the VGG-net [Russakovsky *et al.* (2015)]. The VGG-net architecture (1) is characterized by stacking multiple convolutional layers, then apply a pooling operation. This is repeated multiple times up to one or more fully connected layers, which can serve as task specific head Simonyan and Zisserman (2014).
The latent representations from different convolutional layers are extracted by Gatys *et al.*

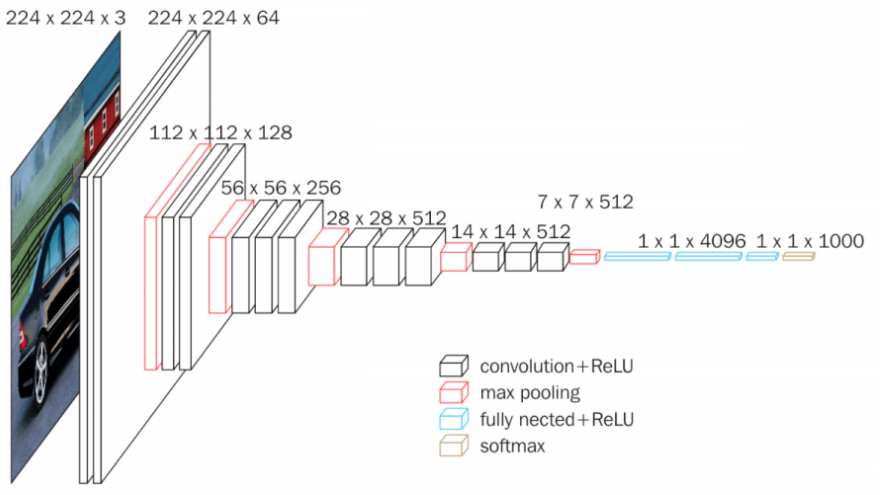

Figure 1: Architecture of the VGG net introduced by Simonyan and Zisserman (2014). Gatys *et al.* (2015) extract feature representations from different convolutional layers.

(2015) and serve as basis for obtaining summarizing statistics. These statistics are used to transform the input image.

3.3 Algorithm

The algorithm starts by extracting a content representation from the content image and style representation of the style image using the VGG-net. From the image, we want to optimize, we extract content as well as style. Then, the distance between the style representations and the content representations is measured. Each distance is a respective loss which gives us two quantities we can minimize. These two loss terms, style loss and content loss, are each multiplied with a small scalar and then added up, giving us our final loss term. The the gradient of the loss function for the current state of the input image is computed and then backpropagated. However only the input image is adjusted according to the gradient and none of the parameters of the VGG-net used to extract the representations.

3.4 Content Representation and Loss

A convolutional layer l with n_l filters returns n_l feature maps which mark the positions of where each respective feature occurred. The height and width of these feature maps is determined by the input and stride used [LeCun *et al.* (1999)]. To extract content information, Gatys *et al.* (2015) propose to gather all feature maps of a layer in a feature Matrix $F_l \in \mathcal{R}^{n_l \times m_l}$ where m_l equal the height times the width of the layers feature maps. Each feature map is vectorized and transposed, resulting in a feature vector and then becomes a

column of the feature matrix. The level of abstraction of features in F_l is sensitive to the choice of l. Gatys *et al.* (2015) propose to use layer that captures the shapes and layout of the input image rather than finer features such as texture information.
To measure the difference between an input image x and content image p, Gatys *et al.* (2015) propose to generate respective feature matrices F^l and P^l and measure mean square distance between these:

$$\mathcal{L}_{content}(p, x, l) = \frac{1}{2} \sum_{ij} (F_{ij}^l - P_{ij}^l)^2 \tag{1}$$

3.5 Style Representation and Loss

The style of an image is invariant to the spatial arrangement of its features. Thus, the style representation should be spatially invariant. Since feature maps contain spatial information and therefore cannot be used to represent style, Gatys *et al.* (2015) propose to measure their co-occurrence in an image. For this purpose, they introduce the Gram matrix. The Gram matrix of a set of vectors contains the inner product of all combination of vectors from the set. In the case of Neural Style transfer, the set of vectors is the set of vectorized feature maps of a layer l. By calculating the inner product of each of the feature vectors, all spatial information is removed. Remaining is the information about how often the features appear in the same position. This gives us a spatially invariant representation of style by capturing what features correlate rather than where a feature appears.

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \tag{2}$$

Gatys *et al.* (2015) found that using multiple layers produced visually more appealing results. So in order to measure similarity in style of input image x and style image a, the respective Gram matrices G^l and A^l are computed. The loss term is

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

where

$$E_l = \frac{1}{4n_l^2 m_l^2} \sum_{ij} (G_{ij}^l - A_{ij}^l)^2$$

and w_l weights each layers contribution to the style loss.

3.6 Total Loss and Optimization

The total loss term is composed of the style loss and the content loss, each weighted by a factor.

$$\mathcal{L}_{total}(p, a, x) = \alpha\mathcal{L}_{content}(p, x) + \beta\mathcal{L}_{style}(a, x)$$

where α and β are hyperparameters that determine the style/content trade-off.
In order to compute gradients that can be backpropagated to the input image, style and content loss need to be derived according to the layers activation. For content loss, we can compute the derivative with

$$\frac{\partial\mathcal{L}_{content}}{\partial F_{ij}^l} = \begin{cases} (F^l - P^l)_{ij} & \text{if } F_{ij}^l > 0 \\ 0 & \text{if} F_{ij}^l < 0 \end{cases}$$

where as style loss can be differentiated by

$$\frac{\partial E_l}{\partial F_{ij}^l} = \frac{1}{n_l^2 m_l^2}((F^l)^T(G^l - A^l))_{ij} \text{ if } F_{ij}^l > 0$$
$$0 \text{ if} F_{ij}^l < 0$$

4 Derivations and Alternative Approaches

Despite the field of Neural Style Transfer being rather young, many alternatives, derivations and improvements to Gatys *et al.* (2015) algorithm have been proposed. Generally, we can differ between two approaches. Image-based approaches iteratively optimize an input image to match the source images content and style. This, however, requires a new training process for every new image. The many resulting backward are computationally expensive. Instead of adjusting an input image, **Model-based approaches** optimize a generative a model that creates stylized images. Thus shifting the computational cost towards training stage and making real-time generation of stylized images possible. In only a few years, many ideas for both approaches have been proposed. In the following section we will have a look at some of those ideas and briefly discuss their benefits and drawbacks.

References

Carroll, N. (2004). Art and human nature. *The journal of aesthetics and art criticism*, **62**(2), 95–107.

Dosovitskiy, A. and Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In *Advances in neural information processing systems*, pages 658–666.

Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Heeger, D. J. and Bergen, J. R. (1995). Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238.

Hertzmann, A. (2003). A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, (4), 70–81.

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. H. (2001). Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340.

Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., and Song, M. (2019). Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*.

Julesz, B. (1962). Visual pattern discrimination. *IRE transactions on Information Theory*, **8**(2), 84–92.

Kolliopoulos, A. (2005). *Image segmentation for stylized non-photorealistic rendering and animation*. University of Toronto.

Kyprianidis, J. E., Collomosse, J., Wang, T., and Isenberg, T. (2012). State of the" art?: A taxonomy of artistic stylization techniques for images and video. *IEEE transactions on visualization and computer graphics*, **19**(5), 866–885.

LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer.

Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196.

Portilla, J. and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, **40**(1), 49–70.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, **115**(3), 211–252.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE.

Winnemöller, H., Olsen, S. C., and Gooch, B. (2006). Real-time video abstraction. *ACM Transactions On Graphics (TOG)*, **25**(3), 1221–1226.