Neural Style Transfer: Using Convolution to Create Images in the Same Artistic Style

Annika Richter, Cornelius Wolff, Juri Moriße

October 30, 2024

Abstract

This paper implements an artistic style transfer approach based on "A Neural Algorithm of Artistic Style" by Gatys et al. (2016). We begin with a brief explanation of how artistic style transfer functions, followed by an overview of various approaches to this challenge. We continue with a more detailed explanation of the pre-trained VGG network and loss functions used in the original paper and in our implementation. Subsequently, we describe our own network implementation of a similar network and present the results obtained with it. Furthermore, we present and compare results obtained from using pre-trained networks and evaluate how different parameters influence the outcome. Additionally, we explore the inverse application of making art more realistic and use our results to highlight the limitations of our approach. Finally, we summarize our findings and outline for future research.

Keywords

Neural Style Transfer, Convolutional Neural Network

1 Introduction

Recreating art can in itself be an artistic process. Stylistic ways to create artistic images can be a unique way that can resemble an artist long throughout their lives. Creating images in a specific artistic style by using a convolutional neural network has been successfully done in the paper "A Neural Algorithm of Artistic Style" by Gatys et al. (2016). Using a pre-trained VGG Network (Simonyan & Zisserman, 2015) their algorithms is capable of creating a new image depicting the content from a content-defining image and the artisitic style of a style-defining image (Gatys et al., 2016). In this paper, we recreated their original algorithm using the pre-trained model and training our own VGG Model. Furthermore, we experimented with different approaches to enhancing the realism in the artistic style, by investigating results that can be approached with different kinds of style and content images.

2 Different Approaches

2.1 Approaches Without Using Neural Networks

To provide a broader perspective on computational style transfer, in the following a brief presentation of two approaches to this task that do not rely on artificial neural networks is given. Those are the stroke-based rendering approach and the region-based abstractions approach. Furthermore an introduction to approaches using neural networks are provided.

2.1.1 Stroke-Based Rendering

One of the first digital approaches to transferring the style of a work of art to an original image was introduced by Haeberli (1990) and is known as Stroke-Based Rendering. This method does not use neural networks for style transfer. Instead, the user paints on a digital canvas that matches the size of the original image. However, the user can only control the size of the stroke, not its color. The orientation of the stroke is determined by an edge detection system at the corresponding image location. The color of the brush is determined by an algorithm that aims to recreate the color tone and image content from the original photo (Haeberli, 1990; Hertzmann, 2003). This results in a highly abstracted version of the photo, as the user can paint in a similar way to classic paintings and thus determine the style, while the actual content is taken from the original photo. In summary, this is not a complete automation of the style transfer, but only a partial automation since the user's input is still required to achieve the target image .

2.1.2 Region-Based Abstractions

Region-based abstractions are another approach to the problem of transferring a certain style of an artwork to a different picture (Semmo et al., 2017). With this approach, the image is divided into different regions depending on color, contours or content (Mould, 2013; Semmo et al., 2017). The size of the different regions can vary significantly. For example, the sky can be grouped into one region, while individual stones or leaves can also represent individual regions. Once the picture has been divided, it can be transformed into a mosaic based on the colors, shapes, etc. in each region (Semmo et al., 2017). A major problem with the approach is that when the picture is automatically divided into regions, errors often occur, resulting in making the style transfer very error-prone. Although there are some techniques, such as smoothing, to limit this problem, there is not yet a generalized working solution to fix this. However, if it is possible to find a well-functioning solution in the future, this could be of great support for creating animations, for example.

2.2 Approaches Using Neural Networks

According to Jing et al. (2020) the main challenge in transferring an artistic style with neural networks is to represent the style of an image. While not completely, a large part of the

information about the style of an image is contained in the information about its texture. This allows to circumvent the problem of computing style representations by using texture representations for which are easier to obtain based on previous work. Therefore, Jing et al. (2020) distinguishes different neural network approaches based on how they represent the texture of an image. They describe two different concepts. The approach used in Gatys et al. (2016) and by us is part of what Jing et al. (2020) call 'Parametric Neural Methods with Summary Statistics. The core idea here is that the texture of an image can be represented by using summary statistics over all the image's pixel values. In our case, the summary statistic in use is the Gram matrix which describes the correlations between filter responses in the VGG networks (Simonyan & Zisserman, 2015). The alternative take on texture representation Jing et al. (2020) call 'Non-Parametric Neural Methods with Markov Random Fields'. Here the texture is not represented based on all pixel values but instead an assumption is made that a certain pixel value is dependent on the pixels around it. Thus, the texture representation is based on an analysis of several smaller neighborhoods of pixels instead of the global analysis over all pixels used in the parametric approach. Jing et al. (2020) point out that parametric approaches are prone to losing spatial information because of their use of global statistics. This leads to poor performance when modeling regular or symmetrical textures as well as for realistic styles. While non-parametric approaches counter those weaknesses, they struggle if the content and style inputs are not similar in shape and perspective (Jing et al., 2020).

3 The VGG Network

In the paper "A Neural Algorithm of Artistic Style," by Gatys et al. (2016), a VGG network consisting of 16 convolutional layers and 5 pooling layers was utilized (Simonyan & Zisserman, 2015). This VGG network is known for its excellent performance in image classification tasks. Its architecture consists of five blocks, each consisting of a pooling layer followed by three convolutional layers (Simonyan & Zisserman, 2015). In the original VGG network, a MaxPooling layer was used, but the authors achieved better results with applying AveragePooling layers instead (Gatys et al., 2016).

Pre-trained versions for both the VGG16 and VGG19 networks are available (Simonyan & Zisserman, 2015). These models can be used directly or fine-tuned for specific tasks. Pre-trained models, such as those trained on the ILSVRC 2012 dataset (commonly known as ImageNet), offer numerous advantages (Deng et al., 2009; Fei-Fei et al., 2009). ImageNet is a vast dataset containing over 15 million labeled images, organized according to the WordNet hierarchy (Deng et al., 2009; Fei-Fei et al., 2009). Training the VGG model on this extensive dataset requires significant time and computational resources (Simonyan & Zisserman, 2015). Utilizing a pre-trained model allows for efficient training with less computational power and time on our loss functions without compromising performance
.

4 Loss Functions

To generate an image that captures both the content of the content input and the style of the style input, gradient descent is applied to a simple noise picture (Gatys et al., 2016). The network aims to find an image that best matches the content features of the content input and the style features of the style input. This is achieved by minimizing two types of loss:

4.1 Content Loss

This loss function aims to preserve the content of the original image in the generated image. It is defined by the squared Euclidean distance between the feature representations of the original image ($\vec{p}$) and the generated image ($\vec{x}$) in a specific layer l:

$$\text{Content Loss} = \|\vec{P}_{ij}^{l} - \vec{F}_{ij}^{l}\|^2$$

where $\vec{P}_{ij}^{l}$ and $\vec{F}_{ij}^{l}$ are the feature representations of the original and generated images in layer l, respectively (Gatys et al., 2016).

4.2 Style Loss

The style loss function aims to transfer the style of the original style image to the generated image (Gatys et al., 2016). It is defined by the mean-square distance between the gram matrices of the two images. Therefore, the gram matrix captures the correlations between feature maps. The differences between the matrix of the style input and the generated image is minimized to ensuring that the generated image matches the style of the style input. In order to do so, the contribution of the style representation of the original and the generated image in each layer are summed up and weighted to get the final loss function (Gatys et al., 2016).

$$\text{Style Loss} = \|\text{Gram}(A) - \text{Gram}(X)\|^2$$

where $\vec{A}$ is the original image and $\vec{X}$ is the generated image. The Gram matrix of an image is defined as:

$$\text{Gram}(A) = \frac{1}{N_l \cdot M_l} \left(\vec{A} \cdot \vec{A}^T\right)$$

where N_l and M_l are the dimensions of the feature maps in layer l.

By combining these two loss functions, the final loss function is designed to ensure that the generated image matches both the content and style of the respective inputs (Gatys et al., 2016).

Figure 1: Style transfer results using the Cifar100 dataset

5 Implementation of VGG13

5.1 Training

Initially, we aimed to implement the VGG network used in the paper by ourselves. However, during the development phase, we decided to use the smaller VGG13 network. This decision was driven by two main advantages:

1. **Reduced Computational Cost**: The VGG13 network significantly reduces the number of parameters to be trained, thereby lowering computational costs.

2. **Limited Computational Resources**: Given the constraints of Google Colab, we opted to train the model using the Cifar100 dataset instead of the larger ImageNet dataset. Cifar100 consists of lower-resolution images and only 100 different classes, making it more manageable for the VGG13 network.

Our training results were consistent with our expectations. After 30 epochs, the model achieved an accuracy of approximately 99.95%. To enhance learning behavior, we reduced the learning rate from 0.001 to 0.0001 after 15 epochs. We used Adam as the optimizer and categorical cross-entropy as the loss function. The batch size was set to 128.

5.2 Style Transfer

During the style transfer process, we encountered the challenge of adapting the model to correctly identify the appropriate content and style layers. This required careful adjustment to ensure that the generated image accurately reflects both the content and style characteristics of the original images.

For style transfer purposes, the model is defined within the constructor of the VGG13 network but is not compiled. Depending on the intended use, one can either compile the model for training or import pre-trained weights using the `import_weights_for_style_transfer()` function. This function imports the specified weights and selects the correct output layers

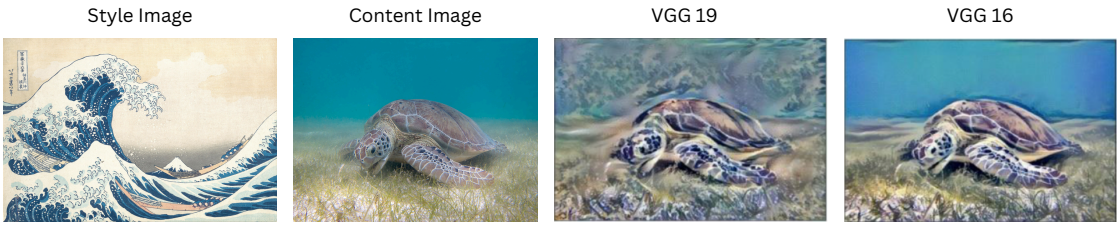

Figure 2: Style transfer results using the pre-trained VGG16 and VGG19 modles

for style transfer. We used 1000 iterations, set the Content Weight to 5000 and the Style Weight to 0.001. The optimizer used was Adam with a learning rate of 5, β_1 set to 0.99, and ϵ set to 0.1. The results are illustrated in Figure 1.

Despite the lower resolution of the Cifar100 dataset images (32x32 pixels), the style transfer was successful and evident in the output image. To achieve higher resolution images, the model would need to be trained using the ImageNet dataset.

6 Using Pre-Trained VGG16 and VGG19

For higher resolution style transfer, we utilized pre-trained VGG16 and VGG19 models. These models allowed us to perform style transfer at a higher resolution, enabling a comparison between the VGG16 and VGG19 models. We used a content weight of 5 and a style weight of 0.001. The network was given 1000 iterations to perform the style transfer, and Adam was used as the optimizer with the same parameters as those used with the VGG13 model. The results are shown in Figure 2.

Using the pre-trained models we were able to make the style transfer in a higher resolution which allowed us to be able to compare the results of using the VGG16 and the larger VGG19 model. The parameters we used were a content weight of 5 and a style weight of 0.001. We gave the network 1000 iterations to perform the style transfer and again used Adam as the optimizer with the same parameters as with the VGG13 model. The results show that the style transfer process profits significantly from using the larger and deeper pre-trained VGG19 model:

6.1 Enhancing Realism in Artistic Styles

After achieving initial results, we explored additional applications. Specifically, we investigated whether swapping the content and style images could produce a more realistic versions of the artistic rendition. To test this idea, we used a photograph of a grazing sea turtle (Lindgren, 2013)as a style image as the style image and Japanese woodcut, 'The Great Wave of Kanagawa,' as the content image (see Figures 3). However, the resulting images (Figure 4) were not more realistic but rather the opposite as they showed a greenish glow presumably coming from the color of the sea turtle image.

To obtain a result with more realistic colors, we replaced the style image with a pho-

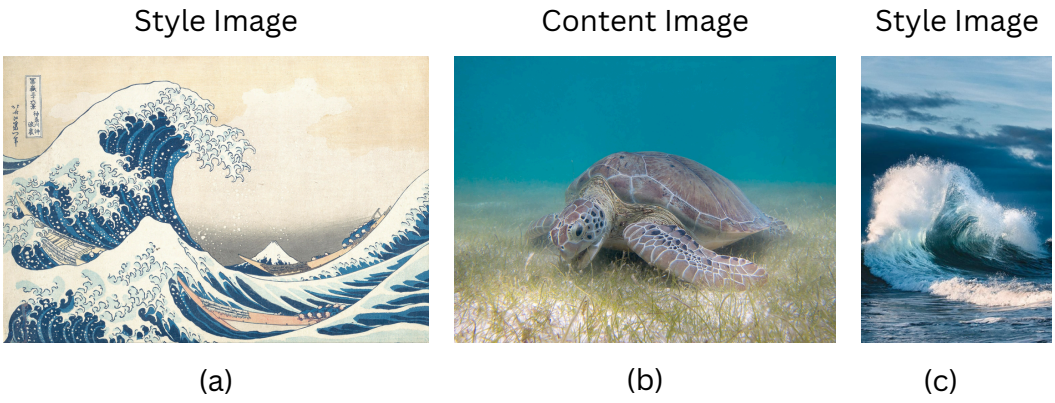

Figure 3: (a) Style Image: Sea Turtle, (b) Content Image: Great Wave of Kanagwa´, (c) Style Image: Realistic Ocean Wave

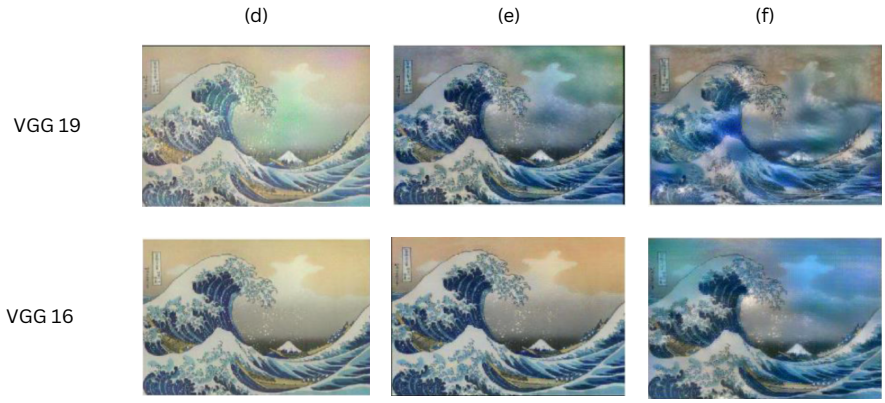

Figure 4: Resulting VGG19 and VGG16 output utilizing on different style and content images.In (d) 'The Great Wave of Kanagawa' was used as the content image. In (e), an image of a realistic ocean wave was used as the style image. In (f), the style weight was further increased(a) Style Image: Sea Turtle, (b) Content Image: Great Wave of Kanagwa´, (c) Style Image: Realistic Ocean Wave

tograph of an ocean wave (Figure 3) (Fenmore, n.d.). This adjustment yielded promising results (Figure 4), shifting the colors to more natural blues and softening the sharp contours typical of woodcuts. However, we were still unsatisfied with the outcome, as the images appeared somewhat graphic and not entirely realistic.

Further refinement was achieved by increasing the style weight. This adjustment led to the disappearance of the boats (when using VGG19) while preserving the waves, showcasing more realistic colors, textures, and blurred contours (Figure 4).

We also applied this method to a self-portrait of Van Gogh as the content image and a portrait photograph as the style image (Figures 4) (Gogh, 1889; Samuelson, 2016). While the results showed more realistic skin tones and a reduction in Van Gogh's characteristic swirls (Figure 4), it did not work as well as with The Great Wave of Kanagawa (Hokusai, 1830). There are several reasons why this inverse application of our approach shows only

Figure 5: Results of using more similar style and content images from the VGG19 and VGG16

limited success in both cases. One problem is that it relies on the image classification network. When using an artistic image as the content image, the underlying network needs to compute a content representation and for that it needs to identify objects in the content image. Because the networks we used are trained on photos, it is understandable that they struggle with classifying objects that are shown in an artistic style.

This is most likely the reason for the disappearance of the boats seen in (Figure 3). A second problem was already outlined previously in the overview of approaches using neural networks. According to Jing et al. (2020) approaches using global statistics like the gram matrix to represent style struggle with regular texture and symmetry in the style picture. This could explain why the result for the Great Wave off Kanagawa was more realistic. The picture of the wave shows very irregular texture and no symmetry while the texture of the portrait photograph used for the Van Gogh portrait is regular and contains symmetrical structures. Additionally, Jing et al. (2020) state that the use of global statistics is generally not suitable for realistic styles. It would be interesting to see if an alternative non-parametric approach using Markov random fields would perform better in this inverse application.

7 Conclusion

In summary, we have succeeded in re-implementing the paper described at the beginning and also succeeded on generating our own results in the form of a new style and content image and the inverse use of style and content images. Furthermore, the implementation of our own VGG13 network serves as a possible foundation for conducting further experiments with the network later on and observing the resulting consequences for the style transfer. However, much larger computing resources will be needed for these experiments, as training on the 150GB ImageNet is likely to be required for this purpose.

References

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255. https://doi.org/10.1109/cvpr.2009.5206848

Fei-Fei, L., Deng, J., & Li, K. (2009). Imagenet: Constructing a large-scale image database. *Journal of vision*, *9*(8), 1037–1037. https://doi.org/10.1167/9.8.1037

Fenmore, J. (n.d.). Ocean waves photography [Last accessed on April 4, 2021]. https://www.pinterest.com.au/pin/624944885772338137/

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2414–2423. https://doi.org/10.1109/cvpr.2016.265

Gogh, V. V. (1889). Self-portrait [Last accessed on April 4, 2021]. https://de.wikipedia.org/wiki/Datei:Self-Portrait_(Van_Gogh_September_1889).jpg

Haeberli, P. (1990). Paint by numbers: Abstract image representations. *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*. https://api.semanticscholar.org/CorpusID:5108976

Hertzmann, A. (2003). A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, *23*, 70–81. https://api.semanticscholar.org/CorpusID:18606937

Hokusai. (1830). The great wave off kanagawa [Last accessed on April 4, 2021]. https://en.wikipedia.org/wiki/The_Great_Wave_off_Kanagawa#/media/File:Tsunami_by_hokusai_19th_century.jpg

Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., & Song, M. (2020). Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*, *26*(11), 3365–3385. https://doi.org/10.1109/TVCG.2019.2921336

Lindgren. (2013). Green sea turtle grazing seagrass [Last accessed on April 4, 2021]. https://commons.wikimedia.org/wiki/File:Green_Sea_Turtle_grazing_seagrass.jpg

Mould, D. (2013). Region-based abstraction. *Image and Video-Based Artistic Stylisation*. https://api.semanticscholar.org/CorpusID:30016828

Samuelson, P. (2016). Portrait photograph of van gogh lookalike [Last accessed on April 4, 2021]. https://images.app.goo.gl/vGmcuyZ25XBddu9y7

Semmo, A., Isenberg, T., & Döllner, J. (2017). Neural style transfer: A paradigm shift for image-based artistic rendering? *International Symposium on Non-Photorealistic Animation and Rendering*. https://api.semanticscholar.org/CorpusID:33492895

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556