Artistic Style Transfer: A Replication Study

J. Linkemeyer, M. Wodrich

April 4, 2021

Contents

1 Introduction

Artistic style transfer aims at converting the content of a given image into the style of a given painting. In 2015, Gatys et al. [1] proposed a deep neural network (DNN) to perform artistic style transform on any given image. Not only is this remarkably interesting because it produces great new images, it is also very powerful from a scientific point of view. In art history, borders of art epochs are often not clearly defined. Whether an image belongs to a certain epoch is often decided based on the year it was painted, the content it depicts, the color scheme it uses and the overall style. However, the categorization process is not always straightforward. Generating new images through artistic style transfer allows taking a closer look at different art epochs, painters and their painting style. This gives the opportunity to analyze and compare styles on a different level, namely solely on the style and not bound to the content. Ideally, this can help identifying patterns for different epochs and the correlation between them.

In order to transfer the style of an image, Gatys et al. [1] use a Convolutional Neural Network (CNN), which is a type of DNN that processes an image using convolution. Convolution describes the process of applying a filter kernel to an image, which can be seen as a feature extraction process. A CNN applies one or more distinct filter kernels in each layer to a given image. During the training process of a CNN, the values of those filter kernels, denoted *weights* in the context of neural networks, are adapted in order to learn a specific task, such as, for example, image classification. Gatys et al. [1] found that style and content of an image are separable. The key idea behind their approach is the finding that object information, or content of an image gets increasingly precise along the layers of a CNN. Therefore, the content of an image is 'stored' in higher layers, while the style of an image can be obtained through the structure of an image, which can be represented by the correlation of its filter responses in all layers.

Separating style and content of an image allows to create new images with the content of one image and the style of another. The present work aims at replicating the proposed technique by Gatys et al. [1]. We will investigate and analyze of our approach to ensure a clear separation between the content and the style of an image.

2 Artistic Style Transfer

Artistic style transfer using deep networks has first been introduced by Gatys et al. [1] in 2015. Since then, many researchers conducted various follow-up studies and investigated different mechanisms to access the style and the content of an image. Consequently, various solutions of how to combine the style of an image with the content of another image have been proposed.

This section will first present the concept of artistic style transfer as originally proposed by Gatys and colleagues. Furthermore, related work will be introducted.

2.1 Style Transfer by Gatys, Ecker, and Bethge

In their approach to artistic style transfer, Gatys et al. use the information that for CNNs trained for object recognition the different layers within the image processing hierarchy of a CNN represent distinct levels of abstraction and sub-information of a given image [1]. Their approach is based on the VGG-19 achitecture [2] pretrained on imagenet. Visualizing a reconstructed image from different layers (see Figure 1), the researchers found that higher layers in the network carry information about the content of an image, instead of precise pixel values as early layers. They investigated the reconstruction of an artistic style by creating a feature space consisting of the correlations of different filter responses from different layers. Results showed that each layer contains stylistic information and that artistic styles can be precisely reconstructed from the combination of the filter responses. Finally, the

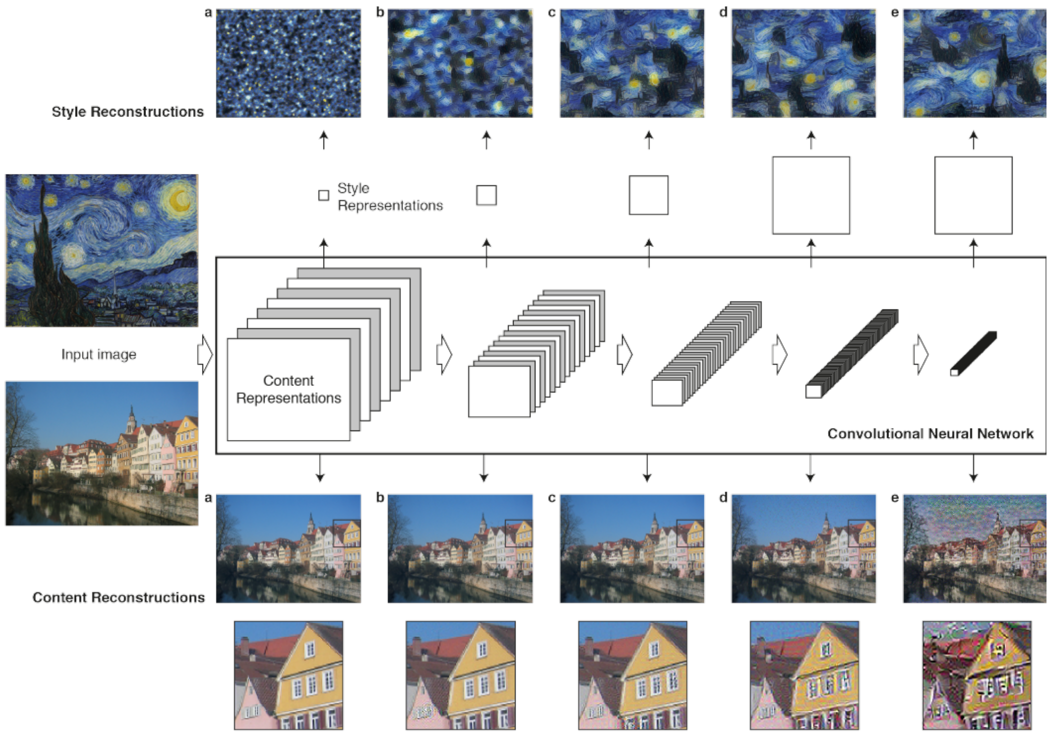

Figure 1: **Style and content reconstructions by Gatys et al.** [3]. The style and content input images have been fed into the CNN to investigate which layers should be used as output layers for the style transfer model. The upper row of images shows the **Style Reconstruction** from different subsets of layers from the VGG-19 model ('conv1 1' (a), 'conv1 1' and 'conv2 1' (b), 'conv1 1', 'conv2 1' and 'conv3 1' (c), 'conv1 1', 'conv2 1', 'conv3 1' and 'conv4 1' (d), 'conv1 1', 'conv2 1', 'conv3 1', 'conv4 1' and 'conv5 1' (e)). The more layers included, the bigger the scale of the style gets. The lower row of images shows the **Content Reconstruction** of the content image using different layer's responses ('conv1 2' (a), 'conv2 2' (b), 'conv3 2' (c), 'conv4 2' (d) and 'conv5 2' (e)). The resulting images indicate that lower the layer, the better the content can be reconstructed.

style representation only consists of texture information regardless of the content and its arrangement.

The researchers use their findings that style and content of an image are separable to create new images capturing the style of one image and the content of another. More precisely, Gatys et al. chose to apply the style of different historically relevant art pieces depicted in Figure 2 to a photograph of the neckarfront (Figure 3).

The resulting image of this style transfer process carries the object information and global arrangement of the content image, combined with the color and local structure of the style image. In a nutshell, the style transfer network allows to apply the styles of impressionism (Vincent van Gogh), expressionism (Edvard Munch), cubism (Pablo Picasso), and various other art epochs to any given input image.

The neural network Gatys et al. introduced takes two images - one content image and

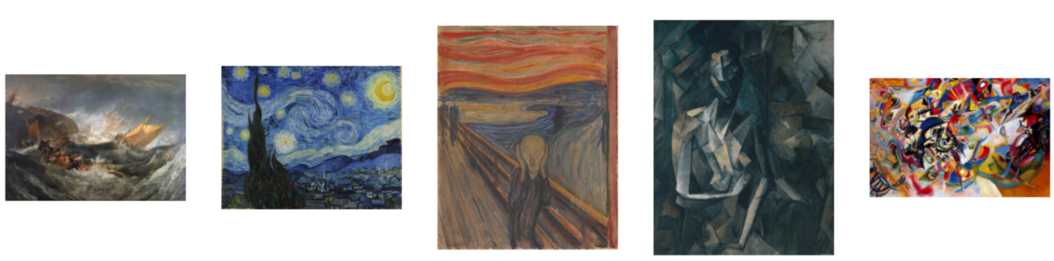

Figure 2: **Art pieces used for artistic style transfer in the paper by Gatys, Ecker and Bethge (2015)** [1]. From left to right: *The Shipwreck of the Minotaur* by J.M.W. Turner (1805), *The Starry Night* by Vincent van Gogh (1889), *The Scream* by Edvard Munch (1893), *Femme nue assise* by Pablo Picasso (1910), and *Composition VII* by Wassily Kandisky (1913).

Figure 3: **Photograph of the Neckarfront in Tübingen, Germany.** This image is used as the content image for the style transfer in the original paper by Gatys, Ecker and Bethge [1].

one style image - as inputs. Because style and content of an image are not well-separable in general, the training of neural network for style transfer aims at finding a trade-off between a loss that is separated into a weighted style and a weighted content loss. For the image the network produces, the feature representation of its content is compared to the one of the content image (denoted in the content loss) using summed squared error. Formally, the content loss is defined as:

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

with $\vec{p}$, $\vec{x}$ being the original and the generated image and F^l, P^l being the feature representations of the images $\vec{p}$, $\vec{x}$ in the layer l.

For the style loss the feature representation of the style is compared to the one of the style image. The calculation here involves the filter outputs from several layers of which we calculate the correlation via a gram matrix. In order to do so, the feature maps from one layer are vectorized and multiplied with each other to get the inner product. Finally, the normalized summed squared error is used to obtain the overall loss of the style. Formally, the entries of the gram matrix G^l can be expressed as:

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

where l denotes the layer and F^l_i, F^l_j represent the vectorised feature maps i and j of the respective layer l. To obtain the style loss of one layer, Gatys et al. compare the gram matrices of the generated image and of the original style image with summed squared error and multiply it with a normalization term. The loss of a style layer l formally can be expressed as:

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

with G^l_{ij} and A^l_{ij} denoting the gram matrices from the generated image and the input style image obtained through the calculation (2). In the normalization term, N_l denotes the number of feature maps in layer l, whereas M_l describes the size of the feature maps of the respective layer, which can be calculated by multiplying its height and width. Finally, the style loss can be obtained by the weighted sum of the losses of all style layers. Formally this is defined as:

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

where w_l denotes the weight with which the respective layer's loss is counted towards the total loss.

To obtain the total loss of the network, both the content loss and the style loss need to be taken into account. This is done by simply weighting and adding them:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style} \tag{5}$$

where α and β are the weights for content and style loss. While this loss is iteratively being improved, the network becomes better at both keeping the content from the original content and in keeping the style from the original style image.

For their style transfer, Gatys et al. chose to use the only specific layers of the pretrained VGG-19 net to train the style and content transfer, as this produced the best results (as indicated in Figure 1). In other words, they build a model which takes the outputs of layer 'conv4_2' (for content) and layers 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1' (for style) as inputs and iteratively adapt the generated image according to the gradients that depend on the weighted content and style losses to perform the task of artistic style transfer. The style loss per layer is weighted by /5, such that each layer counts equally much towards the total style loss.

2.2 Follow-up Studies and Related Work

Following the papers by Gatys et al. [1, 3], several improvements and alternatives have been proposed to perform the task of artistic style transfer. While in the original paper the authors make use of the gram matrix to calculate the style loss, Li et al. [4] combine the idea of using CNNs with Markov Random Fields. Opposed to the the gram matrix which computes a global style, this allows to preserve local patterns of the style inputs which can be used and adapted throughout the whole output image. Later, as an addition to their original work, Gatys et al. [5] proposed several ways on how to achieve a better control mechanisms for the output image, including spatial control, color control and scale control.

As Gatys et al. based their approach on the pretrained object classifier VGG-19, the question arises if maybe other similar networks might perform equally good or even better, ideally being less complex and using less computational resources at the same time. Nikulin et al. [6] compared the performance of VGG-19, VGG-16, AlexNet and GoogLeNet. The results showed that the VGG-19 achitecture is better suited to perform this task compared to the other achitectures, however VGG-16 performs similarly. AlexNet and GoogLeNet lose a lot of fine details, possibly due to large kernels and strides, and therefore perform much worse at the task of artistic style transfer. Additionally, the researchers found that it is possible to only partially transfer the style, meaning that structural and high-level information gets transferred, while the colors from the original image are being kept (see Figure 4). This is achieved via excluding the lower layers from the style loss.

The mechanism behind artistic style transfer proposed by Gatys et al. [1] is an iterative algorithm which can take many iterations, depending on the hyperparameters used, resulting in relatively slow stylization. Johnson et al. [7] optimized the speed by three times via using a different loss approach and combining the idea of feed-forward CNNs with a perceptual loss function. While this speeds up the stylization, it usually is limited to the styles being trained on. In order to tackle this problem, Huang et al. [8] proposed to use an encoder-decoder architecture as a single feed-forward network that works fast is is not bound to being pretrained on a specific style. This allows real time style transfer for any style.

Applying a specific style to a photograph is a task that can be solved with many approaches. While generative adversarial networks (GANs) [9], StyleGANs [10] and Cycle-GANs [11] have shown to produce fascinating results, the present work will focus on the original CNN approach proposed by Gatys et al. [1].

Figure 4: **Style Transfer vs. Partial Style Transfer by Nikulin et al.** Images produced by Nikulin et al. [6] using the upper left image as content image and the upper right image as style image. While normal artistic style transfer uses the colors of the style image (lower left), partial style transfer adapts the structural style of the style image, but uses the colors of the content image.

3 Our Approach

In the present work, we tried to replicate the findings of Gatys et al. [1]. For this purpose, we applied transfer learning and used the VGG-19 architecture [2] pretrained on the ImageNet data set as proposed by the original authors. This network is able to precisely perform image classification and is therefore well suited to extract high-level content information of an image.

The model we built based on the original paper consists of six layers, where one layer is the layer for the content representation and five style layers for the style representation. An analysis for different style layers can be found in the following section, along with choices for other parameters such as content and style weights, amonst others. The model summary can be seen in figure 5. As desired, there are no trainable parameters. The number of non-trainable parameters stems from using a total of six layers from the VGG network - one content and five style layers. In more detail, the following layers were used as in the original paper by Gatys and colleagues:

Content layer: ’block4_conv2’

Style layers: ’block1_conv1’, ’block2_conv1’, ’block3_conv1’, ’block4_conv1’, ’block5_conv1’

Our implementation is available as a Google Colab version, and a local version that can be executed in the console or with a jupyter notebook. Code is available in this github repository.

4 Results and Analysis

We investigated different combinations of content and style images. We first chose to apply our style transfer to the images used in the original paper by Gatys and colleagues. The

```
Model: "style_model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
model (Functional)           ListWrapper([(None, None, 12944960
=================================================================
Total params: 12,944,960
Trainable params: 0
Non-trainable params: 12,944,960
_________________________________________________________________
```

Figure 5: **Summary of our model.**

Figure 6: **Our Results on the Neckarfront image in the styles used in the original paper.** This figure depicts our results using VGG-19's 'block4_conv2' as the content layer and 'block1_conv1', 'block2_conv1', 'block3_conv1', 'block4_conv1', 'block5_conv1' as style layers.

content image is an image of the Neckarfront in Tübingen, Germany, depicted in figure 3. The style images are the artworks presented in figure 2. Figure 6 shows the results on all combinations using our style transfer model. As in the original paper, we chose layer 'block4_conv2' of the VGG-19 model as the content layer and the following layers as style layers: 'block1_conv1', 'block2_conv1', 'block3_conv1', 'block4_conv1', 'block5_conv1'. For the despicted reults, we chose Adam optimizer with a learning rate of 4, content weight 1 and a style weight of 10^{-2}. We optimized the generated image for 2000 epochs. Overall, our results differ from the results by Gatys and colleagues by containing less content of the style image.

To see how the network reacts to other content images, we chose three more content images and performed the artistic style transfer on those as well. For the style images, we chose *The Scream* by Edvard Munch (1893) and *The Starry Night* by Vincent van Gogh (1889) as in the original paper. In addition to that, we also chose the image *Caféterrasse am Abend* by Vincent van Gogh (1888) and an image from the street artist James Rizzy. The original images as well as all possible style transfer combinations are documented in figure 7. We observed that different style images react differently to specific style weights. Using *The Scream* and *The Starry Night* as a style images, the ratio of content and style weight (α/β) was larger compared to the ratio used to transfer to content images to the other two styles. We found that for all variability of the content images - large unicolor areas for the mountain image, the very small details in the sunflower image and the unsharp background of the puppy image - the style transfer works equally well. We also observe that our model captures edges extremely well, especially for the mountain image. For all images, the colors of the style images are captured very well and the brushstroke matches that of the respective artist.

In a further analysis, we captured the effects of different style weights. The results can be found in figure 8. All images were generated using the Neckarfront image as the content image and Kadinsky's *Composition VII* as the style images. For all style transfers, a content weight of 1 was used and the style loss was always lower than the content loss as we observed that this allows for the best results. As expected, for higher style weights, the generated image captures a lot of the style and while the content is still visible, the image is abstracted a lot. We can observe a great difference of the results for style weights between 0.0001 and 0.00001. For the latter, and for style weights even smaller, the ratio of content and style weights gets too large and the content image determines most of the generated image. The

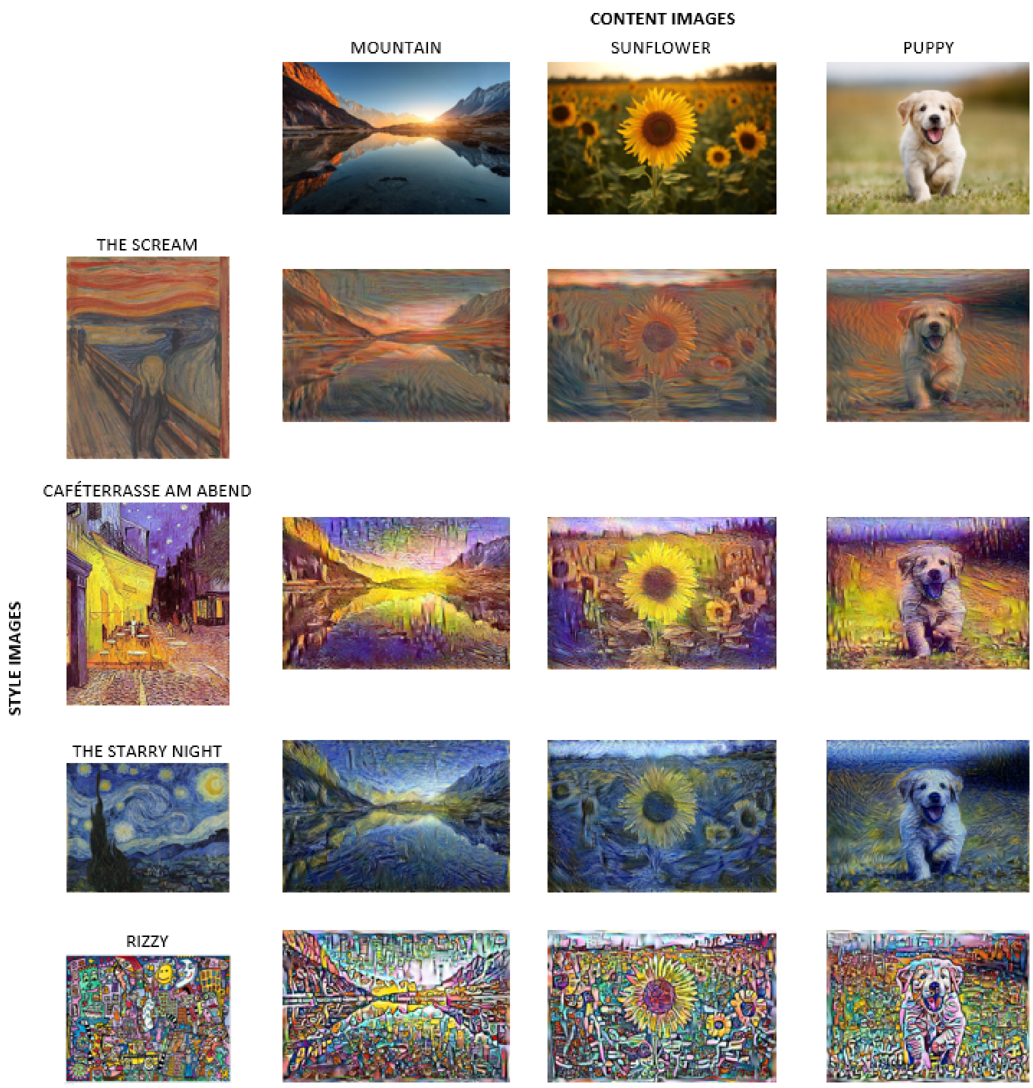

Figure 7: **All combinations of various content and style images.** The content images presented in the first row were obtained through a google search. The style images depicted in the very left column are *The Scream* by Edvard Munch (1893), *Caféterrasse am Abend* by Vincent van Gogh (1888), *The Starry Night* by Vincent van Gogh (1889), and an image from the street artist James Rizzy.

Figure 8: **Analysis of different style weights.** The combination of the Neckarfront image as the content image in the style of Kadinsky's *Composition VII* is examined. All images are created with a content weight $\alpha = 1$, learning rate = 4 and 3000 iterations. The different style weights (β) are documented above each image. It is observable that for high style weights, the image is increasingly abstract and very close to the original style image. For $\beta <= 0.00001$ the content of the Neckarfront image overpowers the style of *Composition VII* extremely.

best results with regard to a trade-off between content and style of the generated image are obtained using a style weight of 10^{-2}, which corresponds to an α/β ratio of 100. This differs from the original paper [1], as the content and style weight ratio to obtain similar results were significantly smaller. Using those ratios, we were unable to get appealing results.

We also conducted experiments using different learning rates and optimizers. We tested our approach using Stochastic Gradient Descent, Adagrad, Adadelta, and Adam. Choosing Adam with a learning rate between one and five yield the best results for our approach. For learning rates smaller than one, we did not get reasonable results. Using other optimizers and adapting the learning rates, content and style weights, we were unable to get similar results compared to the results using Adam. We performed a more detailed loss analysis for the optimizers Adadelta and Adam. The results are shown in Figure 9. We found that for Adadelta, the loss decreases gradually and needs more iterations to produce an appealing generated image. The consistent increase of the content loss for Adadelta might also indicate that using this optimizer, a different content and style weight ratio has to be chosen. For the total loss using the Adam optimizer, a curve with sudden increases that are followed by a further decrease of overall loss than before the increase are observable. This underlines the strength of Adam optimizer to include momentum updates. It is also observable that, using Adam, an overall smaller loss is reached. This explains why the results using Adam are more appealing in the context of artistic style transfer compared to using Adadelta. We therefore consider using Adam optimizer as our default setting and choose a learning rate of 4 for our analyses. The right plot in Figure 9 also allows to conclude that the total loss seems to decrease further after 3500 iterations. Hence, we performed a further analysis on how many iterations our optimization process should be performed.

We tested our approach using Adam optimizer with a learning rate of 4, a content weight of 1, and a style weight of 10^{-2}. We performed the style transfer for 10000 iterations and observed the style and content loss respectively, as well as the total loss. The loss reached its minimum in iteration 8075. Averaging over the losses yielded results that show a minimum loss between 7000 and 8000 epochs. However, by observing intermediate results, we also found that to a human viewer, the generated images do not seem to change drastically after 2000 iterations for most combinations. We therefore use 2000 iterations as our default setting.

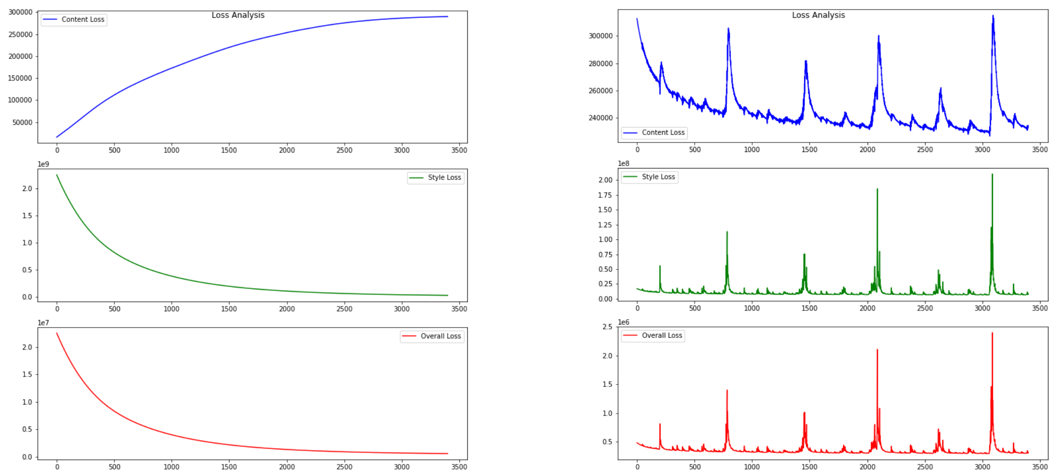

Figure 9: **Loss Analysis for Adadelta and Adam Optimizer.** The left plot shows the losses using Adadelta as the optimizer, the right plot depicts the loss progress when using Adam. Content losses are marked blue (top), style losses green (middle) and the total loss is depicted in red (bottom). Overall, one can observe how the total loss is affected strongly by the style loss. The optimization process was performed for 3500 iterations each.

Additionally, we investigated which combination of style layers is the most appropriate for our desired output. Figure 10 shows that a combination of layers from all convolutional blocks of the VGG-19 network produces the best results. The results also indicate that the lower layers are responsible for the color of the style image, while higher layers are more responsible for texture and structural elements. This is in alignment with the findings from Gatys et al. displayed in figure 1.

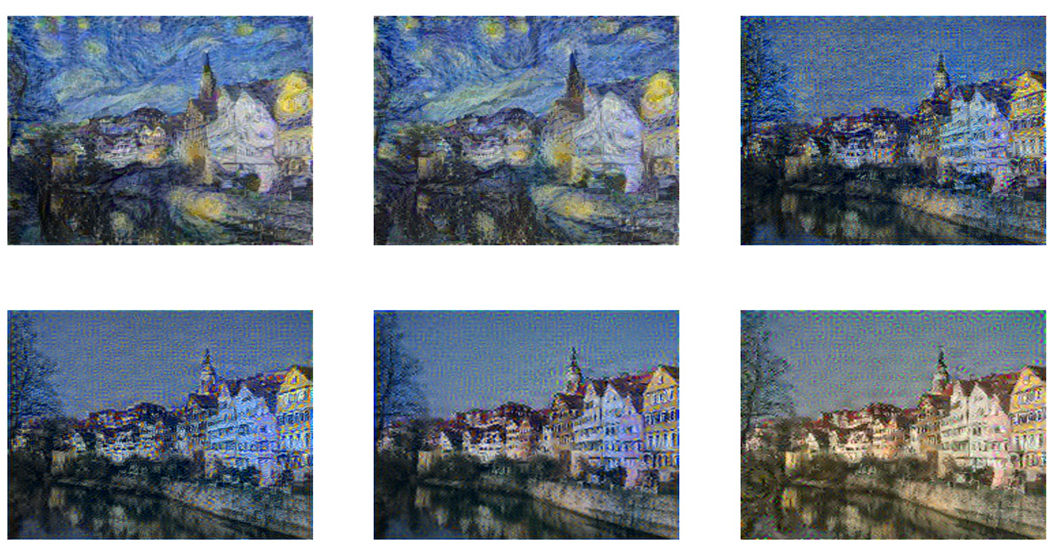

Figure 10: **Analysis of different style layers.** All images were produced with the Neckarfront image and Van Gogh's Starry Night as content and style images. The only difference were the layers used for computing reconstructing the style and calculating the respective loss. Chosen layers were (from left to right, top to down): ('block1_conv1', 'block2_conv1', 'block3_conv1', 'block4_conv1', 'block5_conv1'), ('block1_conv1', 'block2_conv1', 'block3_conv1', 'block4_conv1'), ('block1_conv1', 'block2_conv1', 'block3_conv1'), ('block1_conv1', 'block2_conv1'), ('block1_conv1'), ('block5_conv1').

Nikulin et al. [6] stated that a different assignment of content and style layers yields to partial style transfer that uses the colors from the original image. We were not able to reproduce these findings in an ablation study. With their proposed layers, the generated image still consisted mostly of the colors provided by the style image.

To make further scientific use of style transfer, we additionally analysed various art epochs and compared the resulting images. Figure 11 shows the generated images from four images from cubism and four images from impressionism. All of them were created with the same content images and fixed hyperparameters. To analyse and compare the style, we calculated the style loss between each pair of the produced images. The results show a strong correlation of style for the generated images produced with impressionistic style images. The style loss is a lot higher when comparing these images to the ones that were produced with cubist style images. All impressionistic images show a remarkable difference in style compared to the first (image '0') and also a high one compared to the third (image '3') cubist image. While the style loss between the generated cubist images is still not as low as the style loss in between the impressionistic images, it is still comparatively low. A reason for the higher value could be the different color palettes. Only looking at the colors, this hypothesis seems to hold looking at the heatmap, however further studies would have to be made in order to test this thoroughly. Future studies should also contain a comparison of different style layers used for the calculation of the correlation between the style of two generated images.

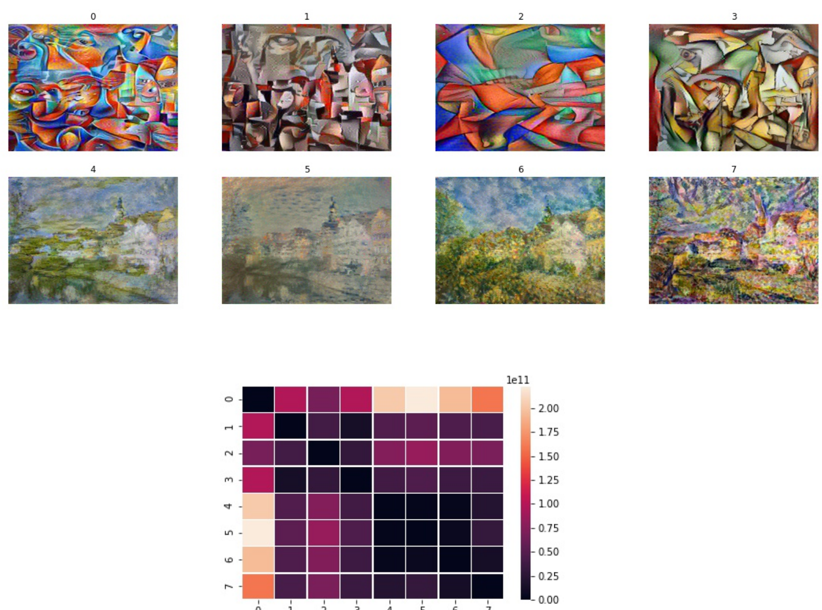

Figure 11: **Analysis of different art epochs.** To compare different art epochs, we performed style transfer using fixed hyperparameters and content images for different style images of two different epochs. The images of the first row were all produced with style images from **cubism**. The images in the second row were created with style images from **impressionism**. The correlations between the different styles are displayed in the **heatmap**. Correlations were computed with the style loss function which compares the styles of two given images. A low value indicates a low loss (similar style), while a high value denotes a big loss, indicating an antagonistic style.

One general drawback of using the CNN approach by Gatys and colleagues is that for each image to be generated, the progress has to be restarted, meaning that we cannot train the model and apply one specific artistic style to as many content images as we prefer at once. This restriction in time and computational resources will be problematic when one wants to produce many generated images. Capturing and storing one specific artistic style is therefore not possible and if it would be, it would be bound to only one style image. A specific style extracted from different images together, like from many paintings from the same artist, is a task yet to be solved with another approach.

5 Conclusion

In conclusion, we produced similar results to those of the original paper by Gatys, Ecker and Bethge [1]. In contrast to our results, the results of Gatys and colleagues seem to contain more content of the style image compared to our results. We were able to get closer to their results by increasing the style weights, however using the original ratios of content and style weight, we could not reconstruct the exact same results. We therefore performed analyses on the parameters we should chose for our model in order to get appealing generated images that accurately capture a trade-off between content of a content image and the style of an artistic style image. It should be noted that this trade-off is subjective to the viewer. Our style weight analysis shows the impact of weight changes, and users can adapt those weights according to their preferences. The additional style layer analysis gives insight into the internal structure and representation of images and their styles. We were therefore also able to compare the style between generated images from different art epochs. This allows to conclude that there are differences in style between different art epochs and that deep learning techniques are a powerful tool to analyze these beyond the visual perception of an observer.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[4] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2479–2486.

[5] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3985–3993.

[6] Y. Nikulin and R. Novak, "Exploring the neural algorithm of artistic style," *arXiv preprint arXiv:1602.07188*, 2016.

[7] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision.* Springer, 2016, pp. 694–711.

[8] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[10] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[11] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

Appendix - Image Sources

In the following we have listed all the sources of the original images we used for creating our images. All Sources were last accessed on April 4, 2021.

Figure	Sources
1	Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge (2015). Reconstructions. https://images2.programmersought.com/382/ae/aed8577a66ab5c333d21aa37027d345e.JPEG
2	William Turner (1810). The Shipwreck of the Minotaur. https://upload.wikimedia.org/wikipedia/commons/2/2e/Shipwreck_turner.jpg Vincent van Gogh (1889). The Starry Night. https://upload.wikimedia.org/wikipedia/commons/thumb/e/ea/Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg/1280px-Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg Edvard Munch (1893). The Scream. https://upload.wikimedia.org/wikipedia/commons/thumb/c/c5/Edvard_Munch%2C_1893%2C_The_Scream%2C_oil%2C_tempera_and_pastel_on_cardboard%2C_91_x_73_cm%2C_National_Gallery_of_Norway.jpg/300px-Edvard_Munch%2C_1893%2C_The_Scream%2C_oil%2C_tempera_and_pastel_on_cardboard%2C_91_x_73_cm%2C_National_Gallery_of_Norway.jpg Pablo Picasso (1910). Femme Nue Assise. https://az334033.vo.msecnd.net/images-7/seated-nude-femme-nue-assise-pablo-picasso-1909-f9095482.jpg Wassily Kadinsky (1913). Composition VII. https://upload.wikimedia.org/wikipedia/commons/b/b4/Vassily_Kandinsky%2C_1913_-_Composition_7.jpg
3	Image of the Neckarfront in Tübingen. https://upload.wikimedia.org/wikipedia/commons/0/00/Tuebingen_Neckarfront.jpg
4	link
5	Own Representation.
6	Image of the Neckarfront in Tübingen. https://upload.wikimedia.org/wikipedia/commons/0/00/Tuebingen_Neckarfront.jpg Wassily Kadinsky (1913). Composition VII. https://upload.wikimedia.org/wikipedia/commons/b/b4/Vassily_Kandinsky%2C_1913_-_Composition_7.jpg Pablo Picasso (1910). Femme Nue Assise. https://az334033.vo.msecnd.net/images-7/seated-nude-femme-nue-assise-pablo-picasso-1909-f9095482.jpg William Turner (1810). The Shipwreck of the Minotaur. https://upload.wikimedia.org/wikipedia/commons/2/2e/Shipwreck_turner.jpg Vincent van Gogh (1889). The Starry Night. https://upload.wikimedia.org/wikipedia/commons/thumb/e/ea/Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg/1280px-Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg Edvard Munch (1893). The Scream. https://upload.wikimedia.org/wikipedia/commons/thumb/c/c5/Edvard_Munch%2C_1893%2C_The_Scream%2C_oil%2C_tempera_and_pastel_on_cardboard%2C_91_x_73_cm%2C_National_Gallery_of_Norway.jpg/300px-Edvard_Munch%2C_1893%2C_The_Scream%2C_oil%2C_tempera_and_pastel_on_cardboard%2C_91_x_73_cm%2C_National_Gallery_of_Norway.jpg
7	A Lake in Front of a Mountain. https://www.scinexx.de/wp-content/uploads/0/1/01-35131-nukliduhr01.jpg

	A Field of Sunflowers. https://www.myhomebook.de/data/uploads/2020/02/gettyimages-1141659565-1024x683.jpg A Puppy on a Field. https://www.mera-petfood.com/files/_processed_/a/4/csm_iStock-521697453_7570f7a9b6.jpg Edvard Munch (1893). The Scream. https://upload.wikimedia.org/wikipedia/commons/thumb/c/c5/Edvard_Munch%2C_1893%2C_The_Scream%2C_oil%2C_tempera_and_pastel_on_cardboard%2C_91_x_73_cm%2C_National_Gallery_of_Norway.jpg/300px-Edvard_Munch%2C_1893%2C_The_Scream%2C_oil%2C_tempera_and_pastel_on_cardboard%2C_91_x_73_cm%2C_National_Gallery_of_Norway.jpg Vincent van Gogh (1888) Cafeterrasse am Abend. https://upload.wikimedia.org/wikipedia/commons/thumb/b/b6/Gogh4.jpg/300px-Gogh4.jpg Vincent van Gogh (1889). The Starry Night. https://upload.wikimedia.org/wikipedia/commons/thumb/e/ea/Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg/1280px-Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg Style from the Artist James Rizzy. http://www.james-rizzi.com/wp-content/uploads/pictures/cache/2007_03_000_KeepingBusyInARizziCity_800_600.jpg
8	Image of the Neckarfront in Tübingen. https://upload.wikimedia.org/wikipedia/commons/0/00/Tuebingen_Neckarfront.jpg Wassily Kadinsky (1913). Composition VII. https://upload.wikimedia.org/wikipedia/commons/b/b4/Vassily_Kandinsky%2C_1913_-_Composition_7.jpg
9	Own Representation.
10	Image of the Neckarfront in Tübingen. https://upload.wikimedia.org/wikipedia/commons/0/00/Tuebingen_Neckarfront.jpg Vincent van Gogh (1889). The Starry Night. https://upload.wikimedia.org/wikipedia/commons/thumb/e/ea/Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg/1280px-Van_Gogh_-_Starry_Night_-_Google_Art_Project.jpg
11	Cubist Image '0'. http://www.boredart.com/wp-content/uploads/2016/08/Excellent-Examples-Of-Cubism-Art-Works-26.jpg Cubist Image '1'. https://www.ygartuaoriginals.com/wp-content/uploads/2017/03/dog-cat-neo-cubism.jpg Cubist Image '2'.http://boredart.com/wp-content/uploads/2016/02/cubism-art-2.jpg Cubist Image '3'. https://tinyurl.com/4y7hn5r6 Impressionistic Image '4'. https://www.livingexcited.com/wp-content/uploads/2018/09/claude_monet_seerosen-_1916-1919_c_fondation_beyeler-_riehen-_basel-sammlung_beyeler.jpg Impressionistic Image '5'. http://www.zeno.org/Kunstwerke.images/I/18v0029a.jpg Impressionistic Image '6'. http://artist.com/art-recognition-and-education/wp-content/themes/artist-blog/media-files/2016/03/impressionism-3.jpg Impressionistic Image '7'. https://www.boredart.com/wp-content/uploads/2015/06/Beautiful-and-Soft-Impressionism-Paintings-71.jpg