Transferring Painting Styles

Cristian Alejandro CHÁVEZ BECERRA, Giovanni BENEDETTI DA ROSA, Paulo Roberto DE MOURA JÚNIOR and Juan Esteban RIOS GALLEGO

Télécom Paris
IMA206: Generative networks, patch methods, computational photography
chavez@telecom-paris.fr
giovanni.benedetti@telecom-paris.fr
paulo.demourajenior@telecom-paris.fr
juan.riosgallego@telecom-paris.fr

Abstract. This project explores the application of neural networks to the artistic task of style transfer, specifically transferring the stylistic elements of renowned artworks to photographic content while retaining the original color integrity of the content images. To perform that, the chosen methodology applies a modified version of Gatys et al.'s neural style transfer algorithm, to selectively transfer texture and color features across different layers. Key adaptations include the use of the ADAM optimizer and adjustments in the loss function ratios to optimize the balance between style and content fidelity. Our methodology extends the traditional style transfer framework by introducing a pre-transfer color alignment process, ensuring the color palette of the style image matches that of the content image and also offering the possibility of transferring colors from another artwork. Through a combination of qualitative and quantitative analyses, including histogram-based color comparisons, we evaluate the effectiveness of our approach. The results demonstrate our method's capability to maintain color accuracy and style detail, providing a robust tool for artistic image transformation.

1 Introduction

The goal of the present work is to implement Gatys et al [1] algorithm for transferring painting styles from a style image, here considered as well-known works of art, to a content image, which is here an arbitrary image. The algorithm uses image representations derived from a Convolutional Neural Network (CNN) pretrained on ImageNet dataset, based on VGGNet architecture.

To transfer the style of an image into the content image, Gatys et al algorithm for neural style transfer [1] was implemented and its behaviour was analyzed. In the analysis, features captured by each of the layers chosen by the author for style loss were tested for different style images and also for different alpha-beta ratios.

In this work, the challenge of color fidelity - a common issue in which the transferred style often overrides the natural color tones of the content image,

leading to less desirable results - has been addressed. By integrating a pre-processing color transferring method, the goal was to ensure that the color palette of the style image harmonizes with that of the content image before style elements are transferred. This is expected to preserve the original colors while still imbuing the stylistic traits from the style image.

Thus, the project's scope includes developing an enhanced version of the Gatys et al. approach for style transfer, optimizing it to maintain color integrity without sacrificing the depth and texture of the style being transferred. As an additional feature, one could transfer only the style of an image while transferring the colors from another image, combining, for example, two works of art into a content image. This involves adjusting hyperparameters that control the influence of CNN layers and the balance between preserving the original image's color scheme and adopting the artistic style. The expected outcome is a robust tool that artists and designers can use to effortlessly combine styles from famous artworks with any photographic content while maintaining the authenticity of the original colors.

2 Methods

2.1 Image Style Transfer Using Convolutional Neural Networks

In the present project the style transfer between images using CNNs is based on Gatys et al. algorithm [1] using the VGGNet-19 architecture, which has 16 convolutional layers grouped in 5 blocks as shown in figure 1. Here the layers $conv1\_1$,$conv2\_1$, $conv3\_1$, $conv4\_1$ and $conv5\_1$ were chosen just like on the original paper.

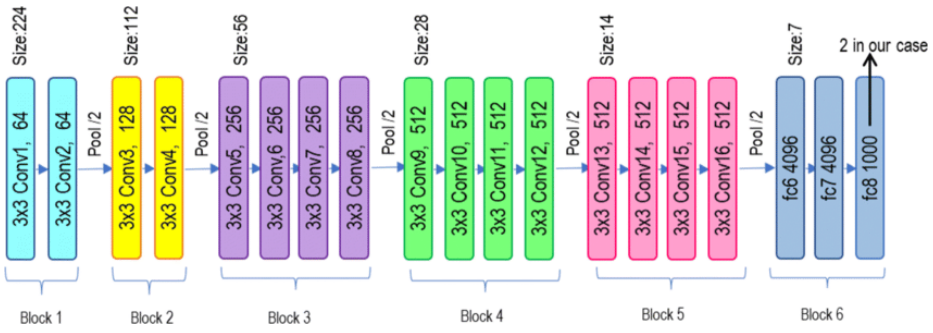

Fig. 1: VGGNet-19 Architecture

Gatys algorithm solves an optimization problem with a loss composed of two components, which will control how much the content image and style image will influence the result. First, a content loss is defined for the layer $conv_1^4$ of the CNN, using an image x, which is the output image of the algorithm containing

the style transfer, and the content image, as defined in the equation below, where F^L and P^L are the gram matrices for image x and content image at layer L respectively.

$$\mathcal{L}_{content} = \frac{1}{|F|} \sum_{i,j} \left(F^L_{i,j} - P^L_{i,j}\right)^2$$

Secondly, a style loss is defined as a weighted sum of multiple losses coming from different layers of the CNN, using the image x and the style image, as described in the equations below, where w_L are the weights, E_L is a single layer loss, G^L and A^L are the gram matrices for image x and style image at layer L respectively. As said before, the layers used to compose this loss in Gatys work are $conv^1_1$, $conv^2_1$, $conv^3_1$, $conv^4_1$ and $conv^5_1$.

$$\mathcal{L}_{style} = \sum_L w_L E_L$$

$$E_L = \frac{1}{|G|} \sum_{i,j} \left(G^L_{i,j} - A^L_{i,j}\right)^2$$

Thus, the optimization problem is defined in the equation below, where α and β control the influence of content and style images in the final result $\hat{x}$.

$$\hat{x} = \arg\min_x \mathcal{L}_{total}$$

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

2.2 Algorithm implementation

The algorithm implemented in this project in Python 3 solves the optimization problem iteratively through the gradient descent method, using Adam optimizer along with a learning rate scheduler both implemented in PyTorch library. In order to compute the gram matrices, the style image is being resized to match the content image, the same approach shown in Gatys work. Furthermore, if the content image has one of its dimensions bigger than 400 pixels, we resize that dimension to 400 pixels, keeping the image ratio, which is done in order to decrease computational time of the algorithm to perform tests but it's not compulsory. The chosen initialization of x is a copy of the content image.

2.3 Qualitative Analysis of layers

In order to evaluate the influence of each layer used in style loss in terms of color and shape patterns in the final result, some tests have been performed fixing the α/β ratio equal to 10^{-7} and the number of iterations equal to 100. To evaluate a single layer influence, the loss E_L was considered as the style loss, thus $\mathcal{L}_{style} = E_L$ varying L. The results for these tests are shown in figure 2.

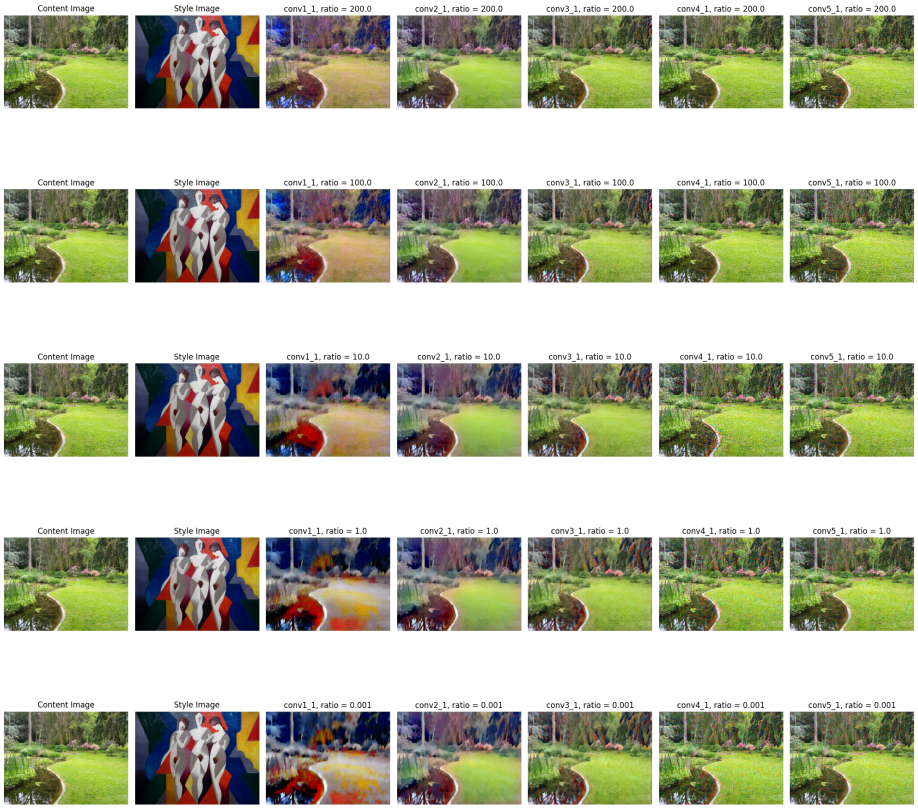

Fig. 2: Comparison of different layers' influence to the final result with $\alpha/\beta = 10^{-7}$ ratio and 100 iterations and varying style images

It's possible to notice on figure 2 that the initial CNN layers $conv_1^1$ and $conv_1^2$ contribute more into transferring the colors from the style images to the final result than the higher layers but they also contribute into transferring the shapes and texture, whereas the higher layers don't transfer color significantly. It should be noted that these results also have influence based on the number of iterations performed and the alpha-beta ratio chosen.

To complement this conclusion, another analysis was performed on the loss function varying α/β ratio in the total loss and using the same style image to see how the result depends on α/β.

It's possible to see in figure 3 that decreasing α/β ratio, giving a higher weight to style loss, the style image will have more influence on the final result, as expected. In addition to that, it's noticeable that, as the ratio decreases, the initial layers warps more the result image and thus it starts to control the shapes and textures instead of just changing the colors. In general, the behaviour of layers is consistent for all alpha-beta ratios which is positive since it's not desired for a hyperparameter to change drastically the behaviour of the algorithm.

2.4 Color Transferring

As an adaptation to the traditional Gatys technique for style transfer, in this work we propose a method to retain the color palette of the content image while still imparting the stylistic elements from the style image. The method consists in adjusting the color palette of the style image to match the colors of the content image and then applying the Gatys method with this transformed image instead of the standard style image.

To perform the color transfer, the task here is make the color statistics of a style image I_S match those of a content image I_C using a linear transformation, before applying Gatys Method.

Given that the transformation involves aligning both the mean and the covariance of the color channels of the content image to have the statistics of the style image, the process is defined by

$$\begin{aligned}
\mu_C &= \frac{1}{N}\sum_{i=1}^{N} I_{Ci} \\
&= \frac{1}{N}\sum_{i=1}^{N} (AI_{Si} + b) \\
&= A\left(\frac{1}{N}\sum_{i=1}^{N} I_{Si}\right) + b \\
&= A\mu_S + b
\end{aligned}$$

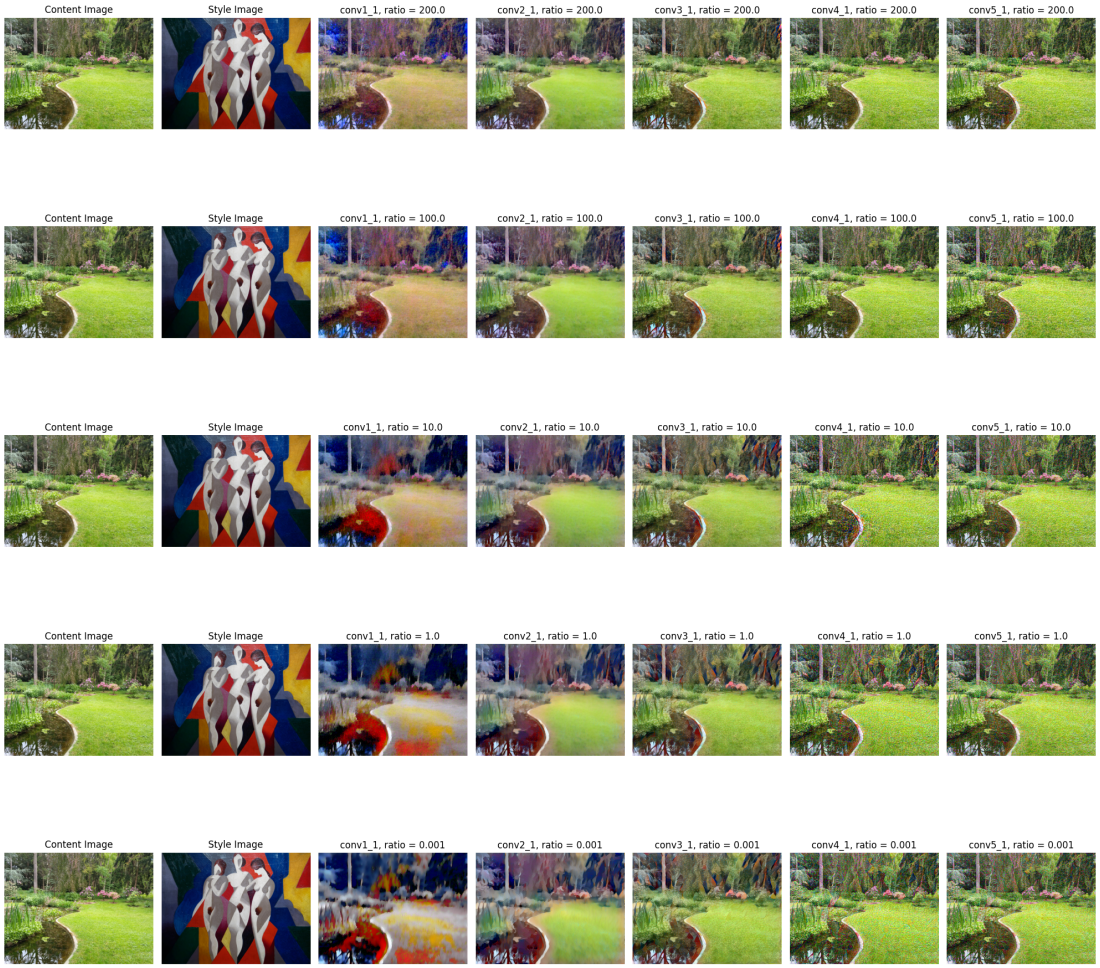

Fig. 3: Comparison of different layers' influence to the final result varying the α/β ratio and 100 iterations with fixed style

$$\begin{aligned}\Sigma_C &= \frac{1}{N}\sum_{i=1}^{N}(I_{Ci}-\mu_C)(I_{Ci}-\mu_C)^T \\ &= \frac{1}{N}\sum_{i=1}^{N}\left[A(I_{Si}-\mu_S)\right]\left[A(I_{Si}-\mu_S)\right]^T \\ &= A\left(\frac{1}{N}\sum_{i=1}^{N}(I_{Si}-\mu_S)(I_{Si}-\mu_S)^T\right)A^T \\ &= A\Sigma_S A^T\end{aligned}$$

where μ_S and μ_C are the means of the style and content images respectively, A is the transformation matrix (3×3), and b (3×1) is the translation vector, finally defining a system of equations

$$\begin{cases}\mu_C = A\mu_S + b \\ \Sigma_C = A\Sigma_S A^T\end{cases} \tag{1}$$

To solve the system of equations 1, the following approach was used: if Σ_C is positive definite, it can be decomposed using the Cholesky decomposition as $\Sigma_C = L_C L_C^T$, where L_C is a lower triangular matrix. Additionally, the eigenvalue decomposition of Σ_S is given by $\Sigma_S = Q\Lambda Q^T$, where Q is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues. Using these decompositions, the matrix A can be determined as $A = L_C Q \Lambda^{-\frac{1}{2}} Q^T$. This model is applicable across various color spaces, however, RGB was chosen here to avoid the inherent angular nature of the a and b components of CIELAB color space and the Hue component of HSV color space, for example. Angular components are a problem because averaging them could lead to color distortions during the transfer process.

In figures 4, 5, and 6 it's possible to visualize the result of the method, as well as the histograms of the respective images. It's thus clear that the distribution of the transformed image is closer to the generated image from the method.

2.5 Quantitative metrics for color transferring

Histogram-Based Image Comparison Color histograms guarantee that the pixels grouped into specific bins will at least be fairly similar in color across different images since they share the same bounds. To quantify the differences between the transformed image and both the content and style images, we employ the Chi-Square distance on their respective color histograms. This method provides a robust way to measure the dissimilarity between two distributions, which in this case are the normalized color histograms of the images.

The RGB color space was chosen to calculate the histograms (and therefore the distances) since we are working in the same space in the proposed color transferring methodology, thus it appeared as a natural choice to compute the comparison.

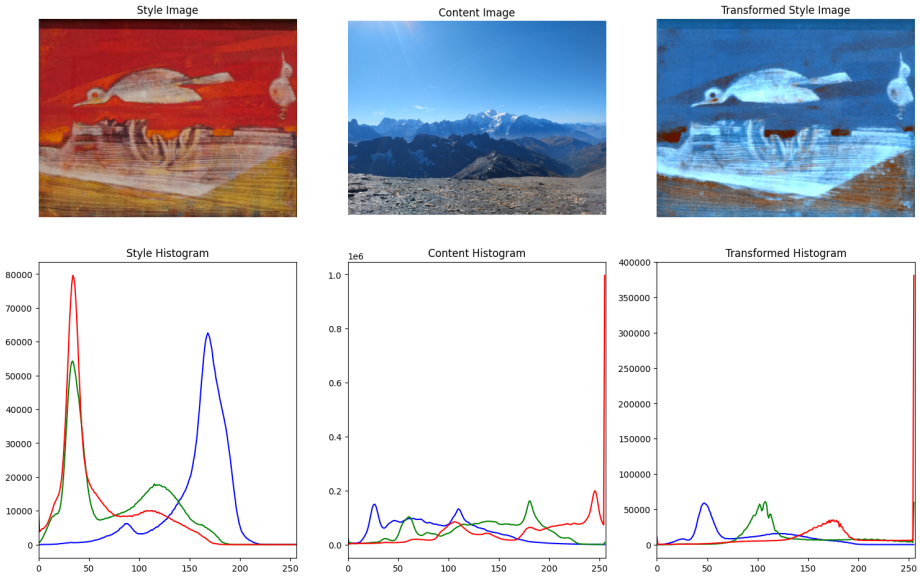

Fig. 4: Results of color transferring method

Color Histogram Computation For each image, the histograms for the Red, Green, and Blue (RGB) channels were computed separately. Let H_R, H_G, and H_B denote the histograms for the Red, Green, and Blue channels, respectively. Each histogram is computed with a fixed number of bins, N, to cover the entire range of possible intensity values (0 to 255).

Thus the color histogram is given by

$$H_c(i) = \text{number of pixels with intensity } i \text{ in channel } c$$

where $c \in \{R, G, B\}$ and i ranges from 0 to 255.

Chi-Square Distance The Chi-Square distance is a statistical measure used to compare two probability distributions. For two histograms, H_A and H_B, the Chi-Square distance can be defined as

$$\chi^2(H_A, H_B) = \sum_{i=1}^{N} \frac{(H_A(i) - H_B(i))^2}{H_A(i)}$$

where $H_A(i)$ and $H_B(i)$ are the counts of the i-th bin in histograms H_A and H_B, respectively. This expression is thus applied to the stacked histograms of two images

$$\chi^2_{total}(H_1, H_2) = \chi^2(H_{1,R}, H_{2,R}) + \chi^2(H_{1,G}, H_{2,G}) + \chi^2(H_{1,B}, H_{2,B})$$

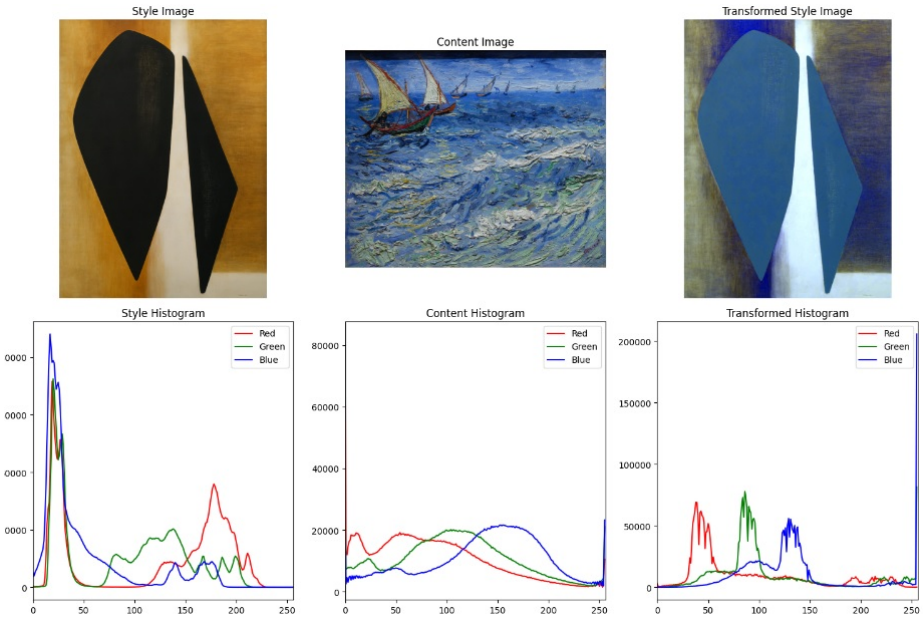

Fig. 5: Results of color transferring method

The chi-squared distance is particularly useful to calculate the difference between two color palettes since it is a weighted sum, therefore, it will the same importance to the different values of each histogram of each channel. Also, this function is symmetric in the x and y axes of the distance making it more suitable for this case where we want to quantify the difference of the colors

Computation with Transformed, Content, and Style Images Let H_T, H_C, and H_S denote the histograms of the transformed image, content image, and style image, respectively. We compute the Chi-Square distances between the transformed image and both the content and style images as follows

$$\chi^2_{TC} = \chi^2_{total}(H_T, H_C)$$

$$\chi^2_{TS} = \chi^2_{total}(H_T, H_S).$$

These distances, χ^2_{TC} and χ^2_{TS}, provide a quantitative measure of the similarity between the transformed image and the content and style images, respectively. A lower Chi-Square distance indicates a higher similarity between the histograms, suggesting that the color distribution of the transformed image is more similar to that of the reference image (content or style). In figures 7, 8 and 9 it's noticeable that the method of color transferring is working well since the Chi-square distance between color and transformed image is always smaller.

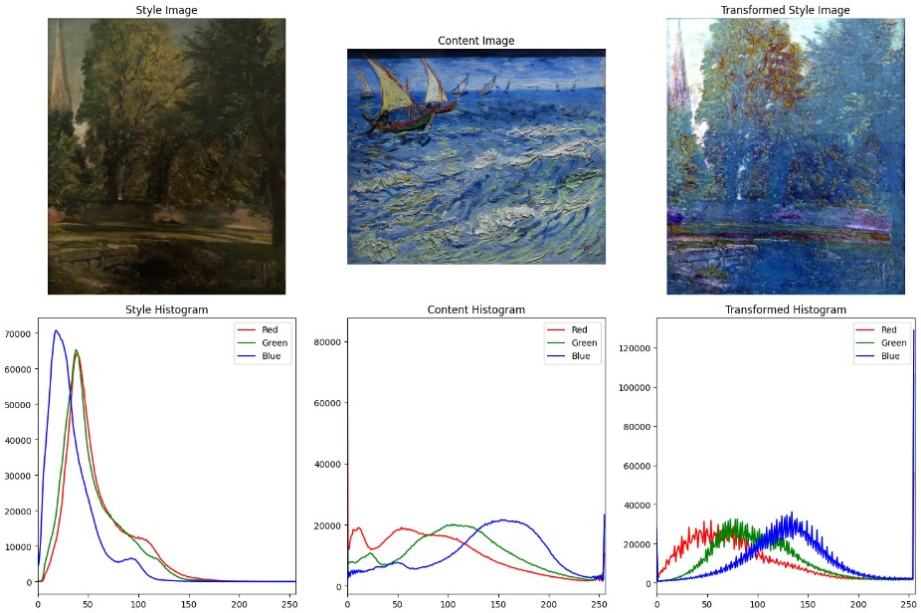

Fig. 6: Results of color transferring method

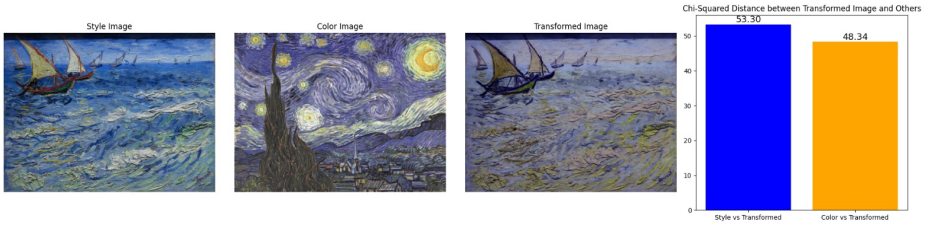

Fig. 7: Distance between images with similar color palettes

Particularly in figure 7 both the style and the color images have similar color palettes, then the metrics reflect subtle differences between the distances calculated between the transformed image with both the style and color images.

3 Results

As it was discussed in the first part of this work, a first approach was made with the implementation of Gatys algorithm where we analyzed mainly the behaviour of the model and its performance, obtaining as first results images with correctly transferred styles. Here the chosen weights to compose style loss were 1, 3/4, 1/5, 1/5, 1/5 for layers *conv*1_1, *conv*2_1, *conv*3_1, *conv*4_1 and *conv*5_1 respectively, and the number of iterations varies from 500 to 4000. In figure 10 it can be

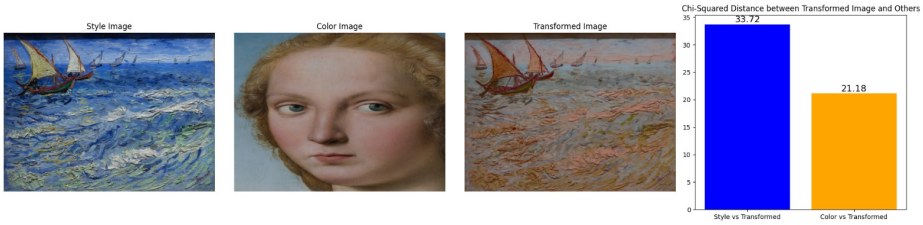

Fig. 8: Distance between images with different color palettes

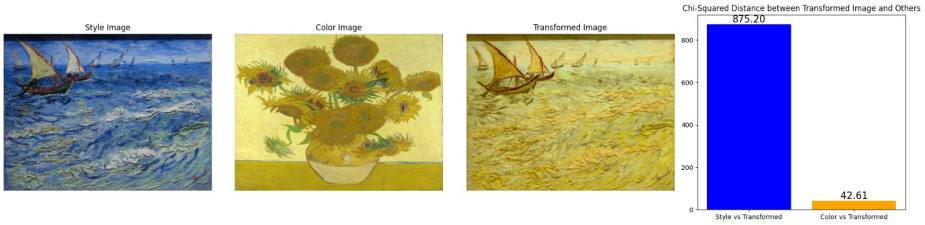

Fig. 9: Distance between images with different color palettes

observed how the synthesized image correctly obtains the style and the textures of the artwork together with the colors of the same with the initial parameters, but also transfer the color palette fro the style image.

Fig. 10: result of a style transfer with gatys algorithm

In a second moment, the project was more focused on adding to the Gatys implementation a way to control the color of the image, using the proposed color transferring method. In this case, the idea was to better control the color of the final image, passing to the style image the original colors of the image to which the style would be transferred.

In a third moment, with a functional color transferring method, the idea was to use a third image, named here "color image" and transfer its color to the style

image, having a transformed image as result and finally transfer the style of this transformed image into the content image.

Figure 11 shows 4 experiments performed that exemplifies what was described here. In the first row, the style image was not transformed using the color image and the output has thus colors of style image.

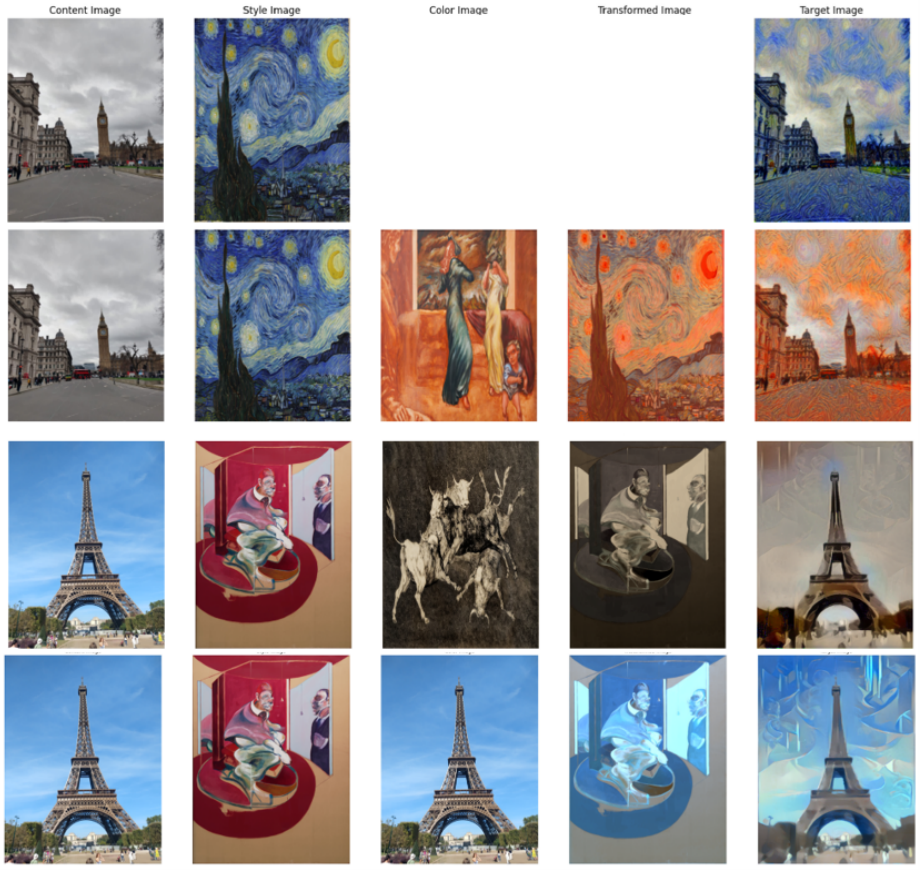

Fig. 11: results of style transfer using different styles, content and color images with Gatys and color transfer method

The second row of figure 11 shows the final image with the style and color of the Van Gogh's art work, but at this time with the color transferred from a painting of Garouste. As a result we can check that despite the fact that the color have changed the style seems similar to the result in the previous row (Gatys Method).

Still in figure 11 we can check two other examples of the Eiffel tower content image with the style transferred from Bacon: in the third row the color's modified

using a Goya painting, while in the fourth row the colors of the content image (Eiffel tower photo) are kept in the final result.

On the other hand, sometimes the method doesn't yield good results as in figure 12. This may be due to several factors, such as the number of iterations performed, an incorrect $\frac{\alpha}{\beta}$ ratio or the weights used in style loss components.

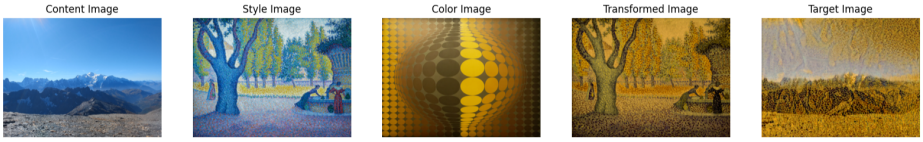

Fig. 12: unsuccessful painting style transfer

4 Conclusions

4.1 Interpretation of results

The results from our style transfer methodology demonstrate a significant retention of the content image color palette while successfully integrating the artistic styles from various well-known artworks. This aligns with the expected outcomes derived from the hypothesis that pre-transfer color alignment would prevent the style's inherent color scheme from overwhelming the content image. Thus, when compared to traditional methods such as those solely based on Gatys et al.'s approach, our method shows a unique capability in maintaining color integrity, which is often not prioritized in conventional style transfer research. In addition to this, one has also the possibility of taking another artwork image and transfer its color to the content image with the color transfer method proved to be functional here.

4.2 Limitations of the method

While our method excels in maintaining color integrity, it can sometimes result in less pronounced style features, particularly in highly textured or complex styles. This is because the preservation of original colors might limit the extent to which style features can be emphasized without altering the underlying color palette.

Moreover, our approach relies heavily on the initial settings of network parameters and the choice of layers within the VGGNet-19 architecture. While we have found a set of parameters that generally perform well, these settings may not be optimal for all types of images or styles. The empirical nature of these settings means that for each new style or content image, significant testing and adjustment may be necessary, which can be a time-consuming and resource-intensive process.

In conclusion, while our approach to neural style transfer introduces significant improvements in color fidelity, its application is best suited for scenarios where color preservation is more critical than the extremity of style transformation. Future work could focus on optimizing these limitations, perhaps by developing more adaptive algorithms that can automatically adjust parameters based on the content and style images, reducing the need for manual calibration and expanding the practical usability of our method.

4.3 Drawbacks of the method

As drawbacks of the method, one should know that:

- Depending on the size of input images it takes a long time to transfer style with enough iterations, thus the method cannot be used as an online algorithm.
- The results of the different tests show that the alpha-beta ratio should be adjusted for every set of input images used.
- The color transferring method used is known to have problems with complementary colors, and thus in some specific cases, it can lead to bad results in color transferring.

4.4 Future Work

In future work, one could consider:

- The possibility of combining different works of art of the same author, to transfer a general style of an artist and not specifically the style of a style image.
- Apply different and more complex color transfer models, using different metrics to measure the results.
- Implement the style transfer algorithm with different CNN models and different optimizers instead of VGGNet-19 with Adam optimizer, in order to speed up the algorithm and possibly getting better results.

References

1. L. A. Gatys, A. S. Ecker and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2414-2423, doi: 10.1109/CVPR.2016.265.
2. H. Weller, "Color Metrics," accessed June 20, 2024. [Online]. Available: https://hiweller.github.io/colordistance/color-metrics.html
3. A. K. Singh and S. K. Dubey, "An Efficient Similarity Measure for Color-Based Image Retrieval," ResearchGate, 2019. [Online]. Available: https://www.researchgate.net/publication/336710156 An Efficient Similarity Measure for Color-Based Image Retrieval
4. M. Safjan, "Metrics to Compare Histograms," accessed June 20, 2024. [Online]. Available: https://safjan.com/metrics-to-compare-histograms/
5. Suvoo, "Image Style Transfer Using CNNs," GitHub repository, accessed June 20, 2024. [Online]. Available: https://github.com/Suvoo/Image-Style-Transfer-Using-CNNs
6. Photonics Media, "Colorimetry: How to Measure Color Differences," accessed June 20, 2024. [Online]. Available: https://www.photonics.com/Articles/Colorimetry How to Measure Color Differences/a25124
7. Lisun Group, "Relationship Between the Use of Colorimeter and Color Space," accessed June 20, 2024. [Online]. Available: https://www.lisungroup.com/news/technology-news/relationship-between-the-use-of-colorimeter-and-color-space.html