



Published in Image Processing On Line on 2024-10-29.
 Submitted on 2024-01-14, accepted on 2024-08-12.
 ISSN 2105-1232 © 2024 IPOL & the authors CC-BY-NC-SA
 This article is available online with supplementary materials,
 software, datasets and online demo at
<https://doi.org/10.5201/ipo1.2024.522>

Non-local Matching of Superpixel-based Deep Features for Color Transfer and Colorization

Roxane Leduc¹, Hernan Carrillo², Nicolas Papadakis³

¹INSA Rouen, France (roxane.leduc@insa-rouen.fr)

²Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, Talence, France
 (hernan.carrillo-lindado@u-bordeaux.fr)

³Univ. Bordeaux, CNRS, INRIA, Bordeaux INP, IMB, UMR 5251, Talence, France
 (nicolas.papadakis@math.u-bordeaux.fr)

Communicated by Lara Raad Demo edited by Lara Raad

Abstract

In this article, we give a thorough description of the algorithm proposed in [H. Carrillo, M. Clément and A. Bugeau, Non-local matching of superpixel-based deep features for color transfer, VISAPP, 2022] for color transfer by relying on a robust non-local correspondence between low-level features at high resolution. An adaptation of this method for colorization is also described. We highlight the overall relevant results obtained with this technique for both applications and also show its limitations.

Source Code

The source code and documentation for this algorithm are available from the [web page of this article](#)¹. Compilation and usage instruction are included in the `README.txt` file of the archive.

Keywords: superpixels; non-local matching; attention mechanism; color transfer; colorization

1 Introduction

This article deals with both color transfer, the process of changing the color distribution of a target image based on a reference image, and colorization, the process of digitally applying color to grayscale images. One practical application of these techniques is, for example, to provide filmmakers with a quasi-automatic tool that makes their tasks less time-consuming and tedious than today's professional software. While both color transfer and colorization can sometimes suffer from poor spatial and color consistency, the method described in the studied paper [3] addresses these issues by relying on robust non-local matching between low-level features at high resolution. The non-local concept has already

¹<https://doi.org/10.5201/ipo1.2024.522>

been used extensively in computer vision, notably by [2] to improve the performance of digital image denoising methods.

The general idea behind the method proposed by [3] is the following. A superpixel segmentation of the target and reference images is first realized using the SLIC algorithm [1, 5]. Next, deep feature maps, which are abstract and semantic representations of an image obtained using a pre-trained deep neural network, are extracted at a superpixel level. Subsequently, a non-local correspondence between superpixels of both images is established using an attention mechanism on the deep features. Global relationships between superpixels are taken into account thanks to this non-local correspondence step, that does not include any additional training. Once the non-local correspondence is established, the pixel-level colors are transferred using a weighted average that takes into account the previously computed attention map between superpixels. All in all, the unsupervised method of [3] is able to transfer the color of the reference image to the target image, while respecting the structure of the target and maintaining a limited computation time for efficient image or video processing.

In this paper, we propose an online implementation of this color transfer method and an extension to the colorization problem. While converting a color image to a grayscale image is a standard task, the reverse operation is a complex problem since no information about the colors to be added is known a priori. This task is classically performed by users, based on their expertise or artistic experience to add hues to monochromatic images. We show that it is possible to automate this colorization process thanks to recent advances in machine learning and neural networks.

The organization of the paper is as follows. Section 2 gives a detailed explanation of the non-local matching technique proposed in [3]. We then study the application of the method to color transfer in Section 3. The proposed extension to colorization is finally explored in Section 4.

2 Non-local Matching Method

In this section, we present the method proposed in [3] for non-local matching of superpixel-based deep features between two RGB images I_T and I_R . In what follows, we will keep the notations of the reference article: I_T will be the target image and I_R the reference one. We first describe in Section 2.1 the extraction of superpixels features, called super-features. The matching process between super-features is then detailed in Section 2.2.

2.1 Super-features Encoding (SFE)

The encoding of the super-features F of an image I takes place in three stages: 1) superpixel decomposition, 2) extraction of deep features using convolutional neural networks, 3) channel averaging process to obtain super-features (see Algorithm 1).

2.1.1 Superpixel Segmentation

Firstly, two superpixel maps are generated using a superpixel decomposition algorithm on the target and reference images. A superpixel is a group of connected pixels that share common characteristics such as similarity of color or intensity and spatial proximity. Superpixels are commonly used to speed up the execution of image processing algorithms and, in some cases, to improve results.

We use the SLIC algorithm [1, 5] which is an adaptation of the k-means clustering algorithm for the image segmentation purpose. SLIC segments an image into superpixels using both color and spatial position information. It associates neighboring pixels based on their similarity and updates the superpixel centers until convergence is reached. Rather than simply decimating the image to reduce the amount of information, segmenting into superpixels provides a set of regions of interest to process, without reducing the amount of raw information in the image. Figure 1 illustrates this

Algorithm 1: Super-Features Encoding (SFE)

Input: f : Features maps $H \times W \times C$

S : Superpixel map $H \times W$

N : Number of segments in the superpixel map

Result: F : Super-Features

for i from 0 to $N - 1$ **do**

$F_T(i) \leftarrow \text{average_pooling}(f(S(i)));$

$F(i) \leftarrow \text{Stack}(F_T(i));$ // piling up superpixel i in matrix F of size $N \times C$

process. This step produces two superpixels maps S_T and S_R that respectively contain N_T and N_R superpixels. We also denote as P_i^T (resp. P_i^R) the number of pixels contained in the i -th superpixel $S_T(i)$ (resp. $S_R(i)$).



Figure 1: Visualization of a superpixel segmentation S obtained with the SLIC algorithm [1, 5].

2.1.2 Deep Features

Deep learning features are obtained from a pre-trained convolutional network applied on the reference and target images I_R and I_T . We use here the first three layers — *Conv1_2*, *Conv2_2* and *Conv3_4* — of a modified VGG-19 architecture [7] as a feature extractor. These first three layers provide a long range of low-level features that suit diverse types of images.

In each case, after the convolution step and the application of a ReLU function, activations are then batch-normalized before moving on to the next layer. This stabilizes activation values, reduces covariation effects between different activations and improves learning convergence. As mentioned above, the VGG architecture we used is a modified one. Indeed, we decided to replace the max-pooling layers from the baseline VGG-19 architecture by an average pooling (see Section 2.1.3).

This step thus produces $l = 3$ (one for each layer) feature maps f_{Tl} and f_{Rl} , composed of $C = 64, 128$ then 256 channels. The spatial dimension of the feature maps f_{Tl} (resp. f_{Rl}) corresponds to the one of the target image I_T (resp. reference image I_R).

Remark. *This approach takes into account features derived from VGG-19, a pre-trained deep convolutional network particularly effective for processing high-dimensional data such as images or videos. The method can nevertheless handle any other handcrafted or learned features.*

2.1.3 Average Pooling

Finally, we apply average pooling to the deep features using superpixel maps S_T and S_R . The features f_T and f_R of the pixels inside each superpixel are averaged per channel and then stacked in the form of matrices known as *super-features*. We therefore obtain three super-feature maps $F_{Tl} \in \mathbb{R}^{N_T \times C}$,

for the target image, with l index ranging from 1 to 3 corresponding to the information from the first three layers of the VGG-19 convolutional neural network and N_T the number of superpixels of the target image. In the same way, for the reference image, we obtain three super-feature maps $F_{R_l} \in \mathbb{R}^{N_R \times C}$, $l = 1 \dots 3$. The overall super-features encoding is illustrated in Figure 2.

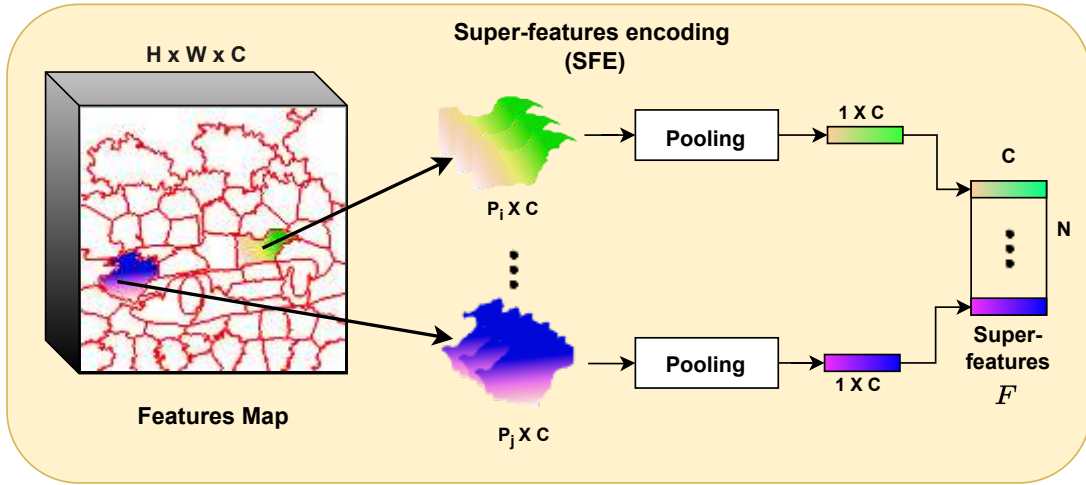


Figure 2: Super-features encoding of [3]. Let H and W be the image size in number of pixels, C the number of channels of the different VGG-19 layers used, N_R (resp. N_T) the total number of reference (resp. target) superpixels, and P_i the number of pixels in the i -th superpixel $S_R(i)$ (resp. $S_T(i)$). The process takes as input a feature map of size $H \times W \times C$, in which each superpixel is extracted and encoded in vectors of size $P_i \times C$. Afterward, the vectors are pooled channel-wise and, finally, stacked in the respective super-features matrices F_{R_l} (resp. F_{T_l}) of size $N_R \times C$ (resp. $N_T \times C$).

2.2 Super-features Matching (SFM)

As stated above, super-features provide a compact encoding for calculating the correlation between deep features. In order to achieve a robust correspondence between the super-features of the target named F_{T_l} and the super-features of the reference named F_{R_l} , the authors were inspired by the attention mechanism described in [9].

Attention mechanisms [8] were popularized with the rise of transformers, a type of neural network architecture that has revolutionized the field of natural language processing and other sequence processing tasks. The attention mechanism allows the model to focus on specific parts of the input when generating the output. Rather than processing the whole sequence at once, transformers use attention to give variable importance to each element of the sequence depending on its context.

The idea of the Super-Features Matching (SFM) process is to exploit the non-local similarities between the super-features of the images by calculating the attention map at layer l as follows

$$A_l(i, \cdot) = \text{softmax} \left(\frac{M_{T_l R_l}(i, \cdot)}{\tau} \right), \quad i = 1 \dots N_T, \quad (1)$$

where, for each superpixel i of the target and j of the reference, we define

$$M_{T_l R_l}(i, j) = \frac{(F_{T_l}(i) - \mu_{T_l})^T (F_{R_l}(j) - \mu_{R_l})}{\|F_{T_l}(i) - \mu_{T_l}\|_2 \|F_{R_l}(j) - \mu_{R_l}\|_2}. \quad (2)$$

In this equation, $M_{T_l R_l}$ corresponds to the correlation matrix between the super-features of the target $F_{T_l}(i)$ and the reference $F_{R_l}(j)$, computed using the mean values $\mu \in \mathbb{R}^C$ over the N_T (resp. N_R) super-feature values. The choice of the normalization through correlation is motivated by the use

of a global temperature parameter $\tau > 0$ to process all components $M_{T_i R_l}$ in the same way. The softmax operation in expression (2) is realized with respect to the second dimension of the matrix, so that $\sum_j A_l(i, j) = 1$ for all superpixels $i = 1 \dots N_T$ of the target image; while $A_l(i, j) \geq 0$ for all $i = 1 \dots N_T$ and $j = 1 \dots N_R$.

The final attention map A is the weighted sum of the attention maps for each layer, divided by the sum of the weights:

$$A(i, j) = \frac{\sum_{l=1}^3 w_l A_l(i, j)}{\sum_{l=1}^3 w_l}, \quad (3)$$

where all weights w_l are set to 1 in our experiments. The value of the attention map $A(i, j)$ can be understood as a measure of the influence of the superpixel $S_R(j)$ of the reference image I_R for the processing of the superpixel $S_T(i)$ of the target image I_T . For further details, please refer to Algorithm 2.

Algorithm 2: Super-Features Matching (SFM)

Input: F_T : Target image Super-features maps $H_T \times W_T \times C$
 F_R : Reference image Super-features maps $H_R \times W_R \times C$
 N_T : Number of segments in the Superpixel map from target
 N_R : Number of segments in the Superpixel map from reference

Result: M_{TR} : Correlation matrix $N_T \times N_R$

for i from 0 to $N_T - 1$ **do**
 for j from 0 to $N_R - 1$ **do**
 $M_{TR}(i, j) \leftarrow \frac{(F_T(i) - \mu_T)^T (F_R(j) - \mu_R)}{\|F_T(i) - \mu_T\|_2 \|F_R(j) - \mu_R\|_2};$

2.3 Complexity Analysis

Let $n = H \times W$ be the number of pixels in an image. Then, the number of features in a deep learning feature map is $D = n \times C$ which translates into a computational complexity of $\mathcal{O}(D^2) = \mathcal{O}(n^2 C^2)$ for computing the correlation matrix. In contrast, with our novel super-features encoding, if we set the number of superpixels in the order of \sqrt{n} , then instead we rewrite with $D_s = \sqrt{n} \times C$, resulting in $\mathcal{O}(D_s^2) = \mathcal{O}(n \times C^2)$ for the correlation operation. As $C \ll n$ can be ignored, we go from a quadratic to a linear complexity operation $\mathcal{O}(n)$.

3 Color Transfer

In this section, we focus on transferring the color of a reference image I_R to a target image I_T . To that end, we follow the color fusion framework initially proposed in [6] which uses the attention maps to obtain the new color for each pixel of the target image.

3.1 Method

Color transfer aims at changing the colors of pixels in the target image I_T using the color palette of the reference image I_R . To that end, the method [3] extends the color fusion framework initially proposed in [6]. The process leverages the attention map A provided by SFE-encoding and SFM-matching, which encodes semantic correspondences between both images. For all pixels inside a superpixel

$S_T(i)$, $i = 1 \cdots N_T$, the attention map $A(i, j)$ is used as a weight to balance the importance of the color to transfer from the reference superpixel $S_R(j)$. For more details please refer to Algorithm 3.

Denoting as $\bar{I}_R(j)$ the mean color value of the pixels belonging to superpixel j of the reference image I_R , the output of the color transfer process on the target image I_T at pixel p is obtained as

$$\hat{I}_T(p) = \frac{\sum_{j=1}^{N_R} W(p, j) \bar{I}_R(j)}{\sum_{j=1}^{N_R} W(p, j)}, \quad (4)$$

where $W(p, j) = \sum_{i=1}^{N_T} d(p, i) A(i, j)$ and $d(p, i)$ depends on the distance between the pixel p and the center $\bar{p}_i = (\sum_{p \in S_T(i)} p) / P_i$ of the i -th superpixel of the target image. This weight is calculated using a Mahalanobis-type formula

$$d(p, i) = e^{-\frac{(V_T(p) - \bar{V}_T(i))^T \Sigma_i^{-1} (V_T(p) - \bar{V}_T(i))}{\sigma_g}} \quad (5)$$

with $V_T(p) = [p, I_T(p)]$ being the vector describing the position and the color of pixel p , and $\bar{V}_T(i) = [\bar{p}_i, \bar{I}_T(i)]$ being the average vector describing the position and color centroids of superpixel $S_T(i)$. The spatial and colorimetric covariances of pixels belonging to the superpixel $S_T(i)$ are computed as

$$\Sigma_i = \begin{pmatrix} \delta_s^2 [Cov(p)]_{p \in S_T(i)} & 0 \\ 0 & \delta_c^2 [Cov(I_T(p))]_{p \in S_T(i)} \end{pmatrix}, \quad (6)$$

where the parameters δ_s^2 and δ_c^2 respectively weight the influence of spatial and color information. The standard deviation σ_g is used to modulate the sensitivity of the $d(p, i)$ distance to the differences between the vectors $V_T(p)$ and $\bar{V}_T(i)$.

In order to optimize computational performance, an initial resizing (*down-sizing*) was applied to the I_R and I_T input images. To guarantee optimum visual quality at the end of the algorithm, we introduce a conversion to the CIELAB color space. We start by converting \hat{I}_T from RGB to CIELAB color space. We thus obtain the 3 channels *Lab* representation $(\hat{L}, \hat{a}, \hat{b})$ for the colorized image \hat{I}_T . We then isolate its chrominance channels \hat{a} and \hat{b} and use them for the resizing operation (*up-sizing*). Next, we concatenate the luminance channel of the original I_T image (also converted into CIELAB space to retrieve the luminance channel) with the resized chrominance channels \hat{a} and \hat{b} of \hat{I}_T . This process replaces the luminance channel of the resulting image \hat{I}_T with the original gray level of I_T . This is an essential step to maintain the structural information present in I_T . Finally, we convert back the resulting image from CIELAB color space to the RGB color space, giving us the final synthesized image. As illustrated in Figure 3, this post-processing makes the colorized images more realistic.



Figure 3: (a) Target; (b) Reference; (c) Resizing without any prior CIELAB conversion; (d) Resizing combined with CIELAB conversion.

In this setting, we achieve a good trade-off between preserving important image information and achieving a compact representation of the image, while having a fast enough process.

Algorithm 3: Color transfer

Input: I_T^0 : Target image $2H_T \times 2W_T \times 3$

I_R^0 : Reference image $2H_R \times 2W_R \times 3$

Result: I_T : Color transferred image $2H_T \times 2W_T \times 3$

$L \leftarrow 3$;

$I_T \leftarrow \text{Resize}(I_T^0, 0.5)$; // Downsize target image with scale factor 0.5. New size: $H_T \times W_T$

$I_R \leftarrow \text{Resize}(I_R^0, 0.5)$; // Downsize reference image with scale factor 0.5. New size: $H_T \times W_T$

$S_T \leftarrow \text{Slic}(I_T)$;

$S_R \leftarrow \text{Slic}(I_R)$;

$N_T, N_R \leftarrow$ Number of superpixels from S_T and S_R ;

for l from 0 to $L - 1$ **do**

$f_{Tl} \leftarrow \text{Conv}_l(I_T)$;

$f_{Rl} \leftarrow \text{Conv}_l(I_R)$;

$F_{Tl} \leftarrow \text{SFE}(f_{Tl}, S_T, N_T, N_R)$;

$F_{Rl} \leftarrow \text{SFE}(f_{Rl}, S_R, N_T, N_R)$;

$M_{TRl} \leftarrow \text{SFM}(F_{Tl}, F_{Rl}, N_T, N_R)$;

for i from 0 to $N_T - 1$ **do**

$A_l(i, :) \leftarrow \text{softmax}\left(\frac{M_{TRl}(i, :)}{\tau}\right)$;

$A \leftarrow \frac{\sum_{l=1}^L w_l A_l}{\sum_{l=1}^L w_l}$;

$H_T \leftarrow$ Height of the target image ;

$W_T \leftarrow$ Width of the target image ;

for h from 0 to $H_T - 1$ **do**

for w from 0 to $W_T - 1$ **do**

$p \leftarrow (h, w)$;

$V_T(p) = [p, I_T(p)]$;

// Position and color of pixel p

for i from 0 to $N_T - 1$ **do**

$\bar{V}_T(i) = [\bar{p}_i, \bar{I}_T(i)]$;

// Average of the position and color centroids of $S_T(i)$.

$\text{dist}(p, i) \leftarrow -(V_T(p) - \bar{V}_T(i))^T \Sigma_i (V_T(p) - \bar{V}_T(i))$; // Σ_i is calculated using Eq. (6)

$d(p, i) \leftarrow e^{-\frac{\text{dist}(p, i)}{\sigma_g}}$;

for j from 0 to $N_R - 1$ **do**

$W(p, j) \leftarrow \sum_{i=1}^{N_T} d(p, i) A(i, j)$;

$\bar{I}_R(j) \leftarrow (1/P_j^R) \sum_{k=1}^{P_j^R} S_R(j)[k]$;

// Mean color of the j -th super pixel of I_R

$\hat{I}_T(p) \leftarrow \frac{\sum_{j=1}^{N_R} W(p, j) \bar{I}_R(j)}{\sum_{j=1}^{N_R} W(p, j)}$;

$\hat{I}_{T_{ab}}(p) \leftarrow \text{RGB2Lab}(\hat{I}_T(p))$;

// Passing from RGB to Lab color space to get ab chrominance

$\hat{I}_{T_{ab}}(p) \leftarrow \text{Resize}(\hat{I}_{T_{ab}}(p), 2)$;

// Upsize ab chrominance with scale factor 2. New size:

$2H_T \times 2W_T$

$I_{T_{gray}} \leftarrow \text{RGB2Gray}(I_T^0)$;

$\hat{I}_T(p) \leftarrow \text{Lab2RGB}(\text{concat}[I_{T_{gray}}, \hat{I}_{T_{ab}}(p)])$; // Concatenation of original luminance and predicted chrominance.

3.2 Experimental Study

Using the online demonstration, the color transfer method [3] can be applied on any pairs of target and reference images. In this section we carry out a series of experiments to describe the influence of the different parameters.

We first run the companion demonstration code on four pairs of real images² to illustrate the pros and cons of this technique applied to color transfer. For this demonstration, and based on empirical testing (see Figure 4), we have set the number of superpixels equal to $3 \times \lfloor \sqrt{H \times W} \rfloor$. Choosing $\lfloor \sqrt{H \times W} \rfloor$ superpixels provides a number of segments proportional to the square root of the total image size. This approach balances detail fidelity (preventing over-segmentation that produces too many superpixels) and computational efficiency (avoiding under-segmentation that loses important details). Figure 4 demonstrates the impact of this hyperparameter on color transfer. In Figures 4(a) and 4(b) the sun’s reflection on the palm tree appears inadequate, whereas this issue is resolved in Figure 4(c). Additionally, the color of the bushes in Figure 4(c) appears more perceptually natural. Multiplying this number by 3 produces the right amount of superpixels that are more accurate at capturing local detail while remaining manageable for subsequent processing algorithms. For more details on the influence of superpixels and an illustration of matched superpixels, we refer the reader to [4].

We now analyze the influence of the different τ , δ_s , and δ_c parameters of the method. As illustrated in Figure 5, the color transfer method can provide plausible results for images containing a single object of interest (flower and bird examples) or more complex scenes (beach example). In the case of the road against a mountain backdrop – see the 4th row of the figure, the correspondence between superpixels is questionable (as an example, the color of the broken line in the middle of the road has remained unchanged).



Figure 4: Influence of number of superpixels.

²Real means no computer-generated images.

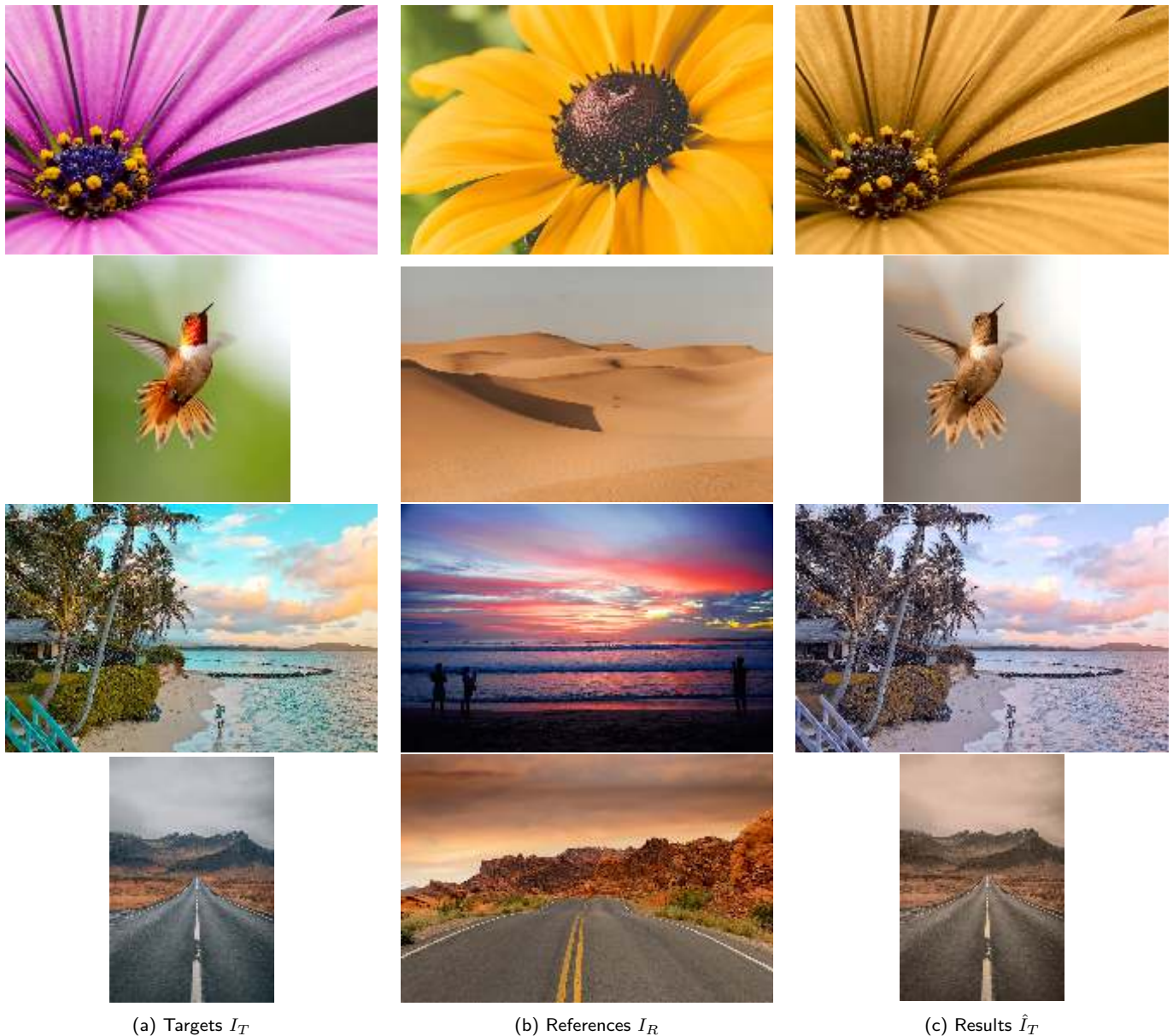


Figure 5: Application of color transfer to 4 different image series.

τ parameter sensitivity analysis. We now evaluate the impact of the main temperature parameter τ of Equation (1), that balances the weights in the attention map A . Figure 6 illustrates the color transfers obtained for the beachfront image and different values of τ . As stated in [3], our experiments suggest that a value $\tau = 1.5 \times 10^{-2}$ gives satisfactory visual results on a large range of images.

As the value of τ increases (for example, $\tau = 1.5 \times 10^{-1}$), the softmax operation of expression (1) makes the probability distribution $A(i, \cdot)$ more uniform. This implies an important mixing of the colors of all superpixels $S_R(j)$, which lead to drab colors in the synthesized image (Figure 6a). On the other hand, when τ decreases, there is a one-to-one matching between a target super-feature and a reference super-feature. For $\tau = 1.5 \times 10^{-4}$, this results in non-uniform color transfer at the bottom left part of the image displayed in Figure 6f.

δ_s and δ_c parameters sensitivity analysis. We now evaluate the impact of the δ_c and δ_s parameters of Equation (6) that weight the influence of color and spatial information. Figure 7 illustrates the color transfers obtained for the road image and different values of δ_s and δ_c . As stated in [3], our experiments suggest that values $\delta_s = 10$ and $\delta_c = 0.1$ give satisfactory visual results on

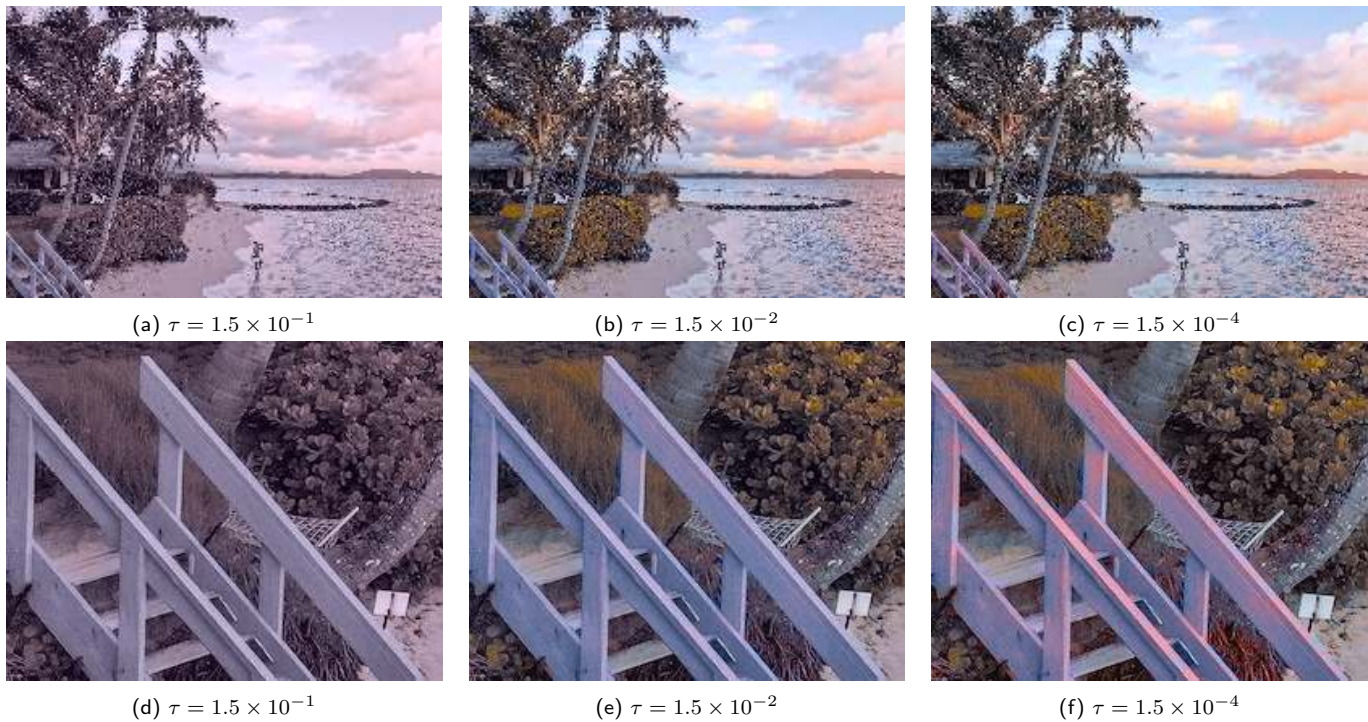


Figure 6: Influence of the temperature value τ on the color transfer.

a large range of images. Indeed, with a preponderance of color information δ_c and less importance given to spatial information δ_s , we observe an inadequate distribution of colors, particularly at the foot of the mountain. In addition, the edges of the road are poorly colored, with dull, lackluster hues that do not faithfully reflect the chromatic palette of the original image.

Limitations. When the shapes and contours of the target image (Figure 8a) differ significantly from those of the reference image (Figure 8b), the transferred colors can appear inappropriate, creating visual artifacts and degrading the perceived quality of the resulting image. In addition, the sharp colors of the reference image may not match the hues of the target image, especially if the latter has more subtle color transitions (see the pixels near the leaves of the tree in Figure 8c). This incompatibility can lead to abrupt and unnatural contrasts in the transformed image, reducing its overall aesthetic.

4 Colorization

We now propose an extension of the method [3] to colorization. In this setting, the target image I_T is a grayscale image, whereas the reference one I_R is a color image.

4.1 Method

The colorization technique we propose consists of 4 steps: 1) transformation of the reference color image into a grayscale one I_{R_g} ; 2) computing SFE-encoding and SFM-correspondences between super-features of grayscale images I_T and I_{R_g} ; 3) synthesis of the colorized target image \hat{I}_T , using attention maps and the original colors of the source image I_R ; 4) post-processing in CIELAB color space, by mixing the original luminance information in the grayscale image I_T with the chrominance channels of \hat{I}_T . For further details, please refer to Algorithm 4.



(a) $\delta_c = 10$ $\delta_s = 0.1$



(b) $\delta_c = 0.1$ $\delta_s = 10$



(c) $\delta_c = 10$ $\delta_s = 0.1$



(d) $\delta_c = 0.1$ $\delta_s = 10$

Figure 7: Influence of the δ_s and δ_c parameters on the color transfer.

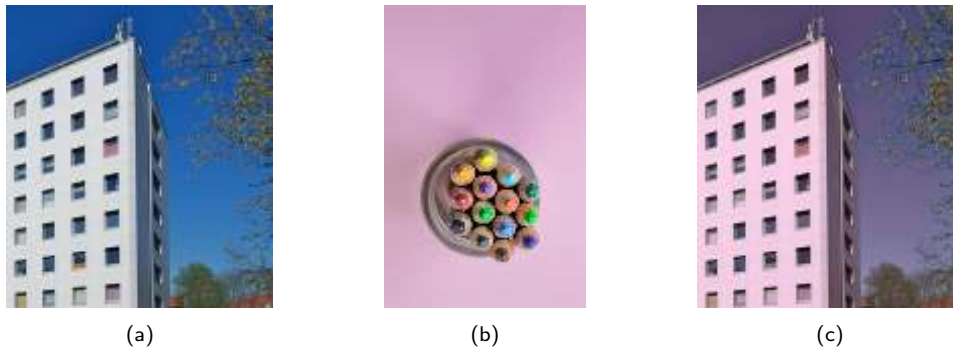


Figure 8: (a) Target; (b) Reference; (c) Result.

From colors to grayscale. The transformation of the RGB color image $I_R = (R, G, B)$ into a grayscale image is done using the standard weighted average from PAL or NTSC models: $I_{R_g} = 0.299R + 0.587G + 0.114B$.

Super-features encoding and matching. We apply the SFE-encoding to both the grayscale image I_{R_g} and the target one I_T . Next we perform the SFM-matching with a slight modification of the attention map computation described in Section 2.2. The merging of the attention maps corresponding to the 3 VGG-19 layers is here realized before the softmax operation:

$$A(i, \cdot) = \text{softmax} \left(\frac{\sum_{l=1}^3 w_l M_{T_l R_{gl}(i, \cdot)}}{\tau} \right), \quad i = 1 \dots N_T. \quad (7)$$

In our experiments, this change appeared useful to avoid the transfer of drab colors.

Color fusion framework. The colorized image \hat{I}_T is obtained by tracing back the original colors of the superpixels in the reference image I_R , as detailed in Equation (4) of Section 3.

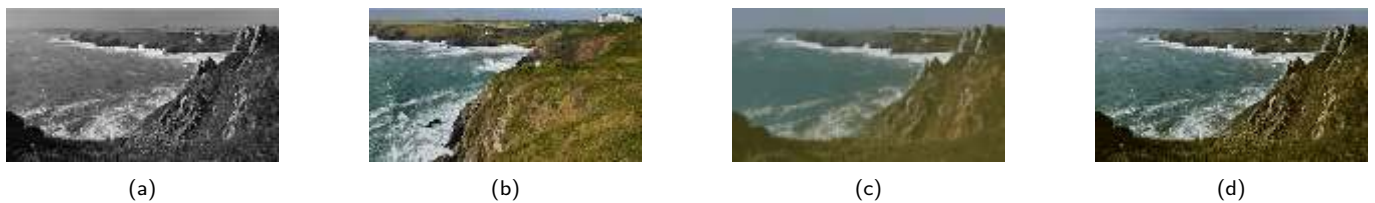


Figure 9: (a) Target; (b) Reference; (c) Resizing without any prior CIELAB conversion; (d) Resizing combined with CIELAB conversion.

Post-processing step. As explained in Section 3, we are going to perform a conversion in CIELAB space in order to improve the visual rendering during resizing, which reduces computation time. We will concatenate the luminance channel of the target grayscale image I_T with the chrominance channel of the resulting image \hat{I}_T . Figure 9 illustrates this process that produces results that are more consistent, more perceptually faithful and less sensitive to variations in brightness.

4.2 Experimental Study

We present in Figure 10 colorization results obtained with the companion demonstration code applied on four real images. When the source and target images are carefully selected, so that they have a particularly high degree of similarity, the visual results are relevant, with realistic colorization (see

Algorithm 4: Colorization

Input: I_T^0 : Grayscale target image $2H_T \times 2W_T \times 1$
 I_R^0 : Reference image $2H_R \times 2W_R \times 3$
Result: I_T : Colorized target image $2H_T \times 2W_T \times 3$
 $L \leftarrow 3$;
 $I_T \leftarrow \text{Resize}(I_T^0, 0.5)$; // Downsize target image with scale factor 0.5. New size: $H_T \times W_T$
 $I_R \leftarrow \text{Resize}(I_R^0, 0.5)$; // Downsize reference image with scale factor 0.5. New size: $H_T \times W_T$
 $I_{R_g} \leftarrow \text{Color_to_grayscale}(I_R)$;
 $S_T \leftarrow \text{Slic}(I_T)$;
 $S_{R_g} \leftarrow \text{Slic}(I_{R_g})$;
 $N_T, N_R \leftarrow$ Number of superpixels from S_T and S_R ;
for l from 0 to $L - 1$ **do**
 $f_{Tl} \leftarrow \text{Conv}_l(I_T)$;
 $f_{Rgl} \leftarrow \text{Conv}_l(I_{R_g})$;
 $F_{Tl} \leftarrow \text{SFE}(f_{Tl}, S_T, N_T, N_R)$;
 $F_{Rgl} \leftarrow \text{SFE}(f_{Rgl}, S_{R_g}, N_T, N_R)$;
 $M_{TRg} \leftarrow \text{SFM}(F_{Tl}, F_{Rgl}, N_T, N_R)$;
 $A_l \leftarrow w \times M_{TRg}$;
for i from 0 to $N_T - 1$ **do**
 $A(i, :) \leftarrow \text{softmax}\left(\frac{\sum_{l=1}^L A_l}{\tau}\right)$;
 $H_T \leftarrow$ Height of the target image ;
 $W_T \leftarrow$ Width of the target image ;
for h from 0 to $H_T - 1$ **do**
 for w from 0 to $W_T - 1$ **do**
 $p \leftarrow (h, w)$;
 $V_T(p) = [p, I_T(p)]$; // Position and color of pixel p
 for i from 0 to $N_T - 1$ **do**
 $\bar{V}_T(i) = [\bar{p}_i, \bar{I}_T(i)]$; // Average of the position and color centroids of $S_T(i)$.
 $\text{dist}(p, i) \leftarrow -(V_T(p) - \bar{V}_T(i))^T \Sigma_i (V_T(p) - \bar{V}_T(i))$;
 $d(p, i) \leftarrow e^{-\frac{\text{dist}(p, i)}{\sigma_g}}$;
 for j from 0 to $N_R - 1$ **do**
 $W(p, j) \leftarrow \sum_{i=1}^{N_T} d(p, i) A(i, j)$;
 $\bar{I}_R(j) \leftarrow$ The mean color value of pixels in superpixel j of I_R ;
 $\hat{I}_T(p) \leftarrow \frac{\sum_{j=1}^{N_R} W(p, j) \bar{I}_R(j)}{\sum_{j=1}^{N_R} W(p, j)}$;
 $\hat{I}_{Tab}(p) \leftarrow \text{RGB2Lab}(\hat{I}_T(p))$; // Passing from RGB to Lab color space to get ab chrominance
 $\hat{I}_{Tab}(p) \leftarrow \text{Resize}(\hat{I}_{Tab}(p), 2)$; // Upsize ab chrominance with scale factor 2. New size:
 $2H_T \times 2W_T$
 $\hat{I}_T(p) \leftarrow \text{Lab2RGB}(\text{concat}[I_T^0, \hat{I}_{Tab}(p)])$; // Concatenation of original luminance and predicted
 chrominance.

for instance the flower and the coastline). The colorization process nevertheless tends to reproduce the most predominant hue in the reference image (dipper and human face examples).

τ parameter sensitivity analysis. We evaluate the influence of the temperature parameter τ

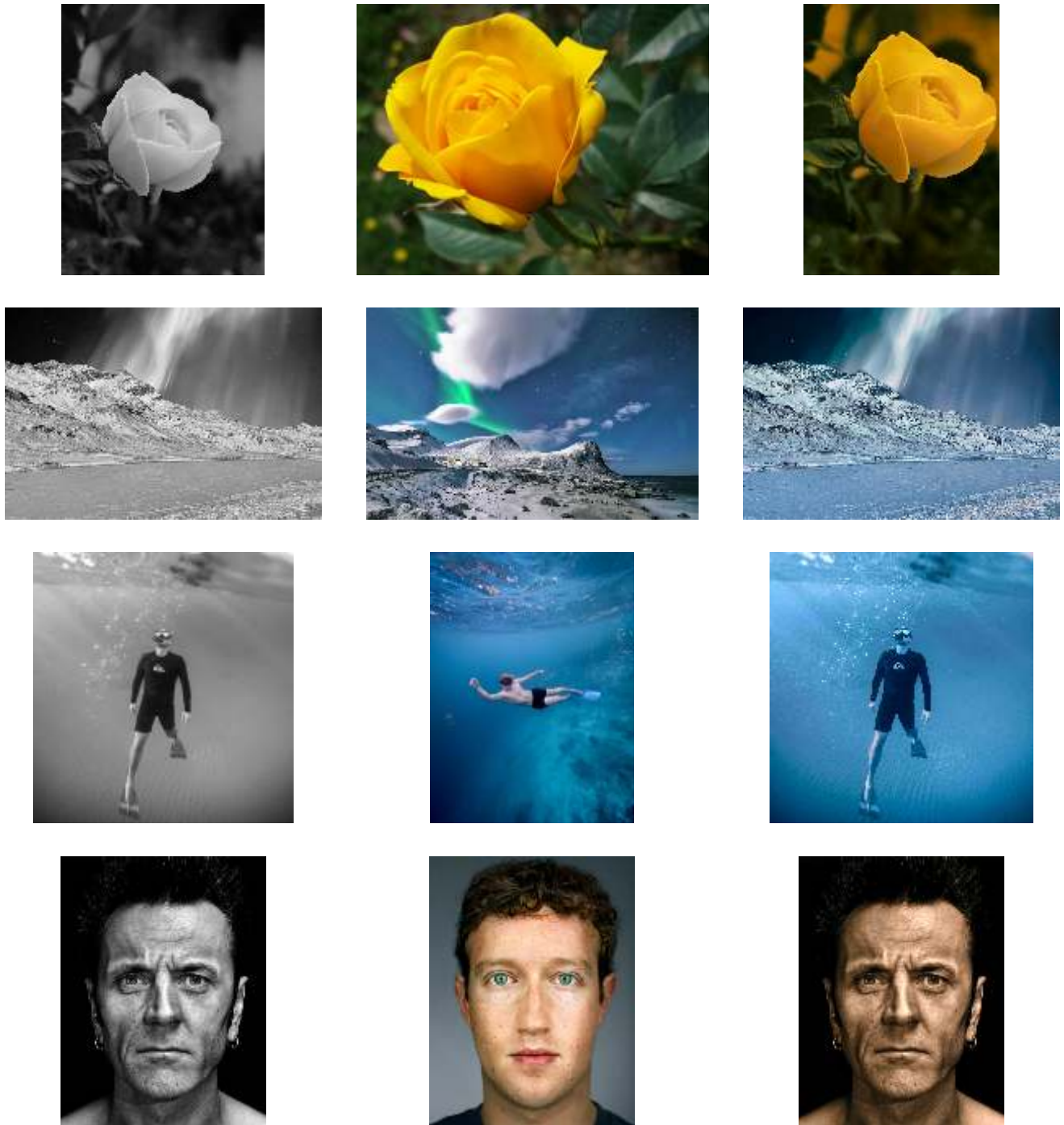


Figure 10: Application of colorization to 4 different image series.

in Equation (7) on the flower example. As illustrated in Figure 11, when $\tau = 1.5 \times 10^{-2}$, the color palette of the reference image I_R is better represented in the colorized image. On the other hand, increasing the value as $\tau = 1.5 \times 10^{-1}$ results in a decrease in the diversity of colors present in the final image.



Figure 11: Influence of the temperature value τ in the colorization results.

δ_s and δ_c parameters sensitivity analysis. We now evaluate the impact of the δ_c and δ_s parameters of Equation (6), that weight the influence of color and spatial information. Figure 12 illustrates the colorization process obtained for the image and different values of δ_s and δ_c . Again, our experiments suggest that values $\delta_s = 10$ and $\delta_c = 0.1$ give satisfactory visual results on a large range of images. Indeed, when the emphasis is placed primarily on color information while neglecting spatial information, the final rendering proves less satisfactory and lacks coherence. Close examination of the image reveals an inconsistent distribution of colors, marked by noticeable and distinct variations in different places. This non-homogeneity creates contrasted areas where hues appear to diverge significantly, introducing visual irregularities within the image.



Figure 12: Influence of the δ_s and δ_c parameters in the colorization results.

Limitations. One of the main problems observed is the tendency to produce an almost monochromatic result, which can appear very artificial. When the reference image (Figure 13b) is dominated by a single hue, this color can be excessively transferred to the target image (Figure 13a), regardless of the latter's natural details. As a result (Figure 13c), complex or varied structures in the target image can be simplified or filled by the predominant blue color of the reference, resulting in a significant loss of chromatic diversity and visual detail. In addition, the marked difference in spatial structures between the two images can lead to inconsistent application of color. For example, a bluish hue from the uniform sky in the reference image may be inappropriately applied to detailed, textured areas of the target image, such as players' shirts. This mismatch creates unnatural visual effects.



Figure 13: (a) Target; (b) Reference; (c) Result.

5 Conclusion

This paper presents a framework for color transfer and colorization of a target image using the color information contained in a reference image. The process is based on non-local matching of deep features extracted from superpixels through an attention mechanism. Experimental results demonstrate the effectiveness of this technique, that manages to preserve the fine details, textures and structures of the target images, while producing consistent and plausible color synthesis using the information from the reference image.

Acknowledgment

This study has been carried out with financial support from the French Research Agency through the PostProdLEAP project (ANR-19-CE23-0027-01).

Image Credits



Jeremy bishop (Pexels)³



Stein Egil Liland (Pexels)³



Frans Van Heerden (Pexels)³



Style transfer for headshot portraits (MIT Open Access Articles)⁴



Style transfer for headshot portraits (MIT Open Access Articles)⁴



Rennon kiefer (Pexels)³



Nikhil Singh Rajput (Pexels)³



Pixabay (Pexels)³



James Lee (Pexels)³



Ylanite Koppens (Pexels)³

³<https://www.pexels.com/fr-fr/discover/>

⁴https://dspace.mit.edu/bitstream/handle/1721.1/100018/Durand_Style.pdf



References

- [1] R. ACHANTA, A. SHAJI, K. SMITH, A. LUCCHI, P. FUA, AND S. SÜSTRUNK, *SLIC Superpixels Compared to State-Of-The-Art Superpixel Methods*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (2012), pp. 2274–2282, <https://doi.org/10.1109/TPAMI.2012.120>.
- [2] A. BUADES, B. COLL, AND J.-M. MOREL, *A Non-Local Algorithm for Image Denoising*, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2005, pp. 60–65, <https://doi.org/10.1109/CVPR.2005.38>.
- [3] H. CARRILLO, M. CLÉMENT, AND A. BUGEAU, *Non-Local Matching of Superpixel-Based Deep Features for Color Transfer*, in International Conference on Computer Vision Theory and Application (VISAPP), 2022, pp. 38–47, <https://doi.org/10.5220/0010767900003124>.
- [4] H. CARRILLO, M. CLÉMENT, AND A. BUGEAU, *Super-Attention for Exemplar-Based Image Colorization*, in Asian Conference on Computer Vision, 2022, pp. 4548–4564, https://doi.org/10.1007/978-3-031-26284-5_39.
- [5] R. GAY, J. LECOUTRE, N. MENOURET, A. MORILLON, AND P. MONASSE, *Bilateral K-Means for Superpixel Computation (the SLIC Method)*, Image Processing On Line, 12 (2022), pp. 72–91, <https://doi.org/10.5201/ipol.2022.373>.

- [6] R. GIRAUD, V.-T. TA, AND N. PAPADAKIS, *Superpixel-Based Color Transfer*, in IEEE International Conference on Image Processing (ICIP), 2017, pp. 700–704, <https://doi.org/10.1109/ICIP.2017.8296371>.
- [7] K. SIMONYAN AND A. ZISSERMAN, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, ArXiv Preprint ArXiv:1409.1556, (2014), <https://doi.org/10.48550/arXiv.1409.1556>.
- [8] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, L. KAISER, AND I. POLOSUKHIN, *Attention Is All You Need*, Advances in Neural Information Processing Systems, 30 (2017), <https://doi.org/10.48550/arXiv.1706.03762>.
- [9] B. ZHANG, M. HE, J. LIAO, P. V. SANDER, L. YUAN, A. BERMAK, AND D. CHEN, *Deep Exemplar-Based Video Colorization*, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 8052–8061, <https://doi.org/10.1109/CVPR.2019.00824>.