Image Style Transfer With a Group of Similar Styles

Valeriy Ponamaryov[1], Victor Kitov[2,3]

[1]Yandex.Technologies LLC, Ulitsa Lva Tolstogo 16, Moscow, 119021, Russia
[2]Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia
[3]Plekhanov Russian University of Economics, Stremyanny lane 36, Moscow, 117997, Russia

Abstract

Image style transfer is a widely used applied task of automatic redrawing of the source image (content) in the style of another image (setting the target style). Traditional style transfer methods provide only a single stylization result. If the user does not like it, for example, due to artifacts that appear during styling, then he has to choose another style. The paper proposes a modification of the styling algorithm, which gives a variety of styling results with one style, and also improves the average quality of styling by using not only style information from the original style image, but also information from images that have a similar style.

Keywords

image generation, image transformation, style transfer diversity, neural networks.

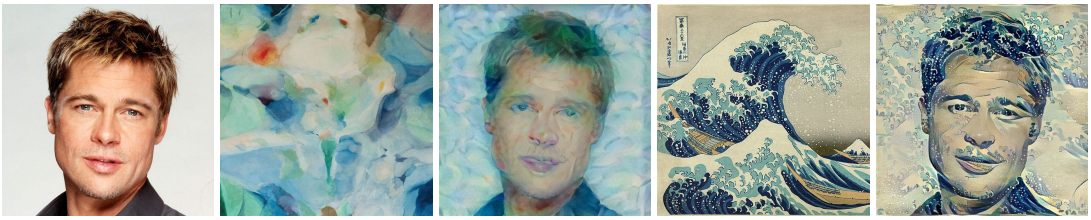
1. Introduction

The algorithm of neural style transfer receives two images as input — a specific image (content) and a style image, representing target style. The algorithm solves the problem of automatically redrawing the content image in the style of the style image. Style refers to the color scheme and characteristic rendering patterns, such as the brush strokes of an artist. This task is relevant for creating vivid illustrations in books, websites, advertising, design, as well as in the entertainment industry.

This task was originally known as non-photorealistic rendering [1, 2, 3] and was solved by heuristic methods of image processing, selected for each style. The neural network style transfer proposed in [4] made it possible to transfer the style from an arbitrary sample style image, as shown in the example of stylizing a content image with two styles in fig. 1.

Fig. 1 shows that the quality of stylization essentially depends on the compatibility of the styles of the content and style images. If they are compatible (style 2), then the styling result is acceptable. If the style of the content and style images differ significantly, for example, in clarity, as when styling with the first style in fig. 1, which is more blurry than the content, then the styling is unsatisfactory. It turns out that in order to create spectacular stylizations, the user is forced to manually sort through various styles for a long time until he finds a style that goes well with the content.

GraphiCon 2023: 33rd International Conference on Computer Graphics and Vision, September 19–21, 2023, V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia
valera.pon.vp@gmail.com (V. Ponamaryov); v.v.kitov@yandex.ru (V. Kitov)
0000-0002-3198-5792 (V. Kitov)



content style 1 stylization 1 style 2 stylization 2

Figure 1: Applying a style transfer method [4].

To simplify the procedure for creating high-quality stylizations, it is proposed to stylize not with one user style, but with a whole set of images similar in style to the input user style image. To calculate stylistic similarity, a special procedure for calculating stylistic characteristics is proposed. Due to the aggregation of style information from several images, the stability of the algorithm to possible individual discrepancies between content and style images is increased. Surveys of respondents show that the proposed modification yields stylizations of better quality.

Another advantage of the proposed modification is the ability to look at the style more broadly. Now it is not set by the source style image alone, but by a whole collection of images that have a style similar to the target one. Thus, by taking different subsets of these images, you can get different styling options. This is useful if the user would like to get a different styling option with the original style.

It should be noted that the basic approach to styling [4], on which the add-on is offered, performs optimization in the original color space of the image being styled, which takes several tens of seconds on the video card and is not applicable, for example, to styling video streams in real time. Many subsequent works, for example [5],[6], were devoted to speeding up styling by transforming the content image through a special neural network. However, this add-on is quite general and applicable to many of them.

2. Base Image Style Transfer Algorithm

Let's consider the original approach specified in the article [4]. The style transfer process operates on the following inputs:

- S – an image containing the desired art style;
- Y is the image whose contents you want to display in the desired style (content).

The task of the styling algorithm is to generate an image X (stylization), in which the content objects Y will be displayed in the style of the style image S. A style defines the color scheme and characteristic patterns of a stylistic image — angles, color changes, characteristic patterns, such as brush strokes of an artist.

To transfer the style, an optimization problem is solved in the pixel space of the resulting stylized image X:

$$\mathcal{L}_{cont}(X, Y) + \alpha \mathcal{L}_{style}(X, S) \to \min_{X}, \tag{1}$$

where $\mathcal{L}_{cont}(X, Y)$ losses penalize the discrepancy between the content image and styling in terms of meaning (what exactly is depicted), and $\mathcal{L}_{style}(X, S)$ penalizes divergence of style image and stylization by style (as shown). The $\alpha > 0$ hyperparameter controls the tension between a more accurate rendering of content (meaning) and a more complete rendering of style.

The content and style losses are calculated based on the representations of X, Y, S images on the intermediate layers of the VGG [7] network, trained to classify images, on the ImageNet [8] sample. Moreover, one intermediate layer k is used to calculate the content loss, and a combination of earlier and later layers is used to calculate the style loss, as shown in Fig. 2.

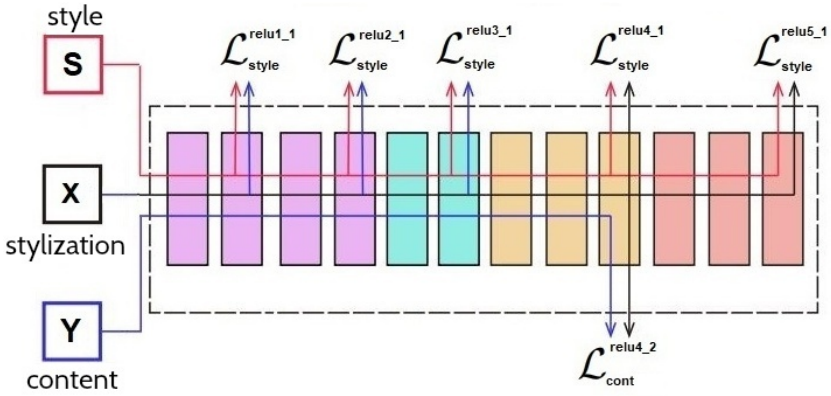


Figure 2: VGG network layers used in styling.

Content loss function:

$$\mathcal{L}_{cont}(X, Y) = \frac{1}{H_k W_k C_k} \sum_{c=1}^{C_k} \sum_{i=1}^{H_k} \sum_{j=1}^{W_k} (X_{cij}^k - Y_{cij}^k)^2,$$

where X^k, $Y^k \in \mathbb{R}^{C_k \times H_k \times W_k}$ are tensors of intermediate representations on some layer k (hyperparameter) of images X and Y , after they were passed through the classification neural network. C_k is the number of channels, and $H_k \times W_k$ is the spatial resolution of the feature map. The essence of the function is that if the images X and Y have different meanings, then their intermediate representations in the VGG network will also differ. Differences are penalized precisely in intermediate feature representations (responsible for semantics), and not in the original RGB representations, since otherwise it would serve as too strong a binding of styling to the original content image and styling would not work.

The style loss is the sum of the style loss for the individual layers of the VGG network:

$$\mathcal{L}_{style} = \sum_k \mathcal{L}_{style}^k,$$

where

$$\mathcal{L}_{style}^k(X, S) = \frac{1}{C_k^2 H_k^2 W_k^2} \sum_{i=1}^{H_k} \sum_{j=1}^{W_k} (G_{ij}^{X^k} - G_{ij}^{S^k})^2. \quad (2)$$

$G^{Z^k} \in \mathbb{R}^{C_k \times C_k}$ denotes the Gram matrix consisting of scalar products between all possible channels of the intermediate representation Z^k in the VGG network on layer k for image Z (equal to either style image S or styling result X):

$$G_{ij}^{Z^k} = \sum_{i=1}^{W_k} \sum_{j=1}^{H_k} Z_{cij}^k Z_{cij}^k, \tag{3}$$

The content determines the spatial arrangement of objects in the image, therefore, when calculating the content loss function, intermediate representations are compared in relation to spatial coordinates (i and j).

Style defines the overall distribution of colors and more general patterns (borders, corners, overflows, brush strokes, etc.). Therefore, this distribution is first extracted in the form of scalar products between channels (when calculating scalar products, spatial information is lost, since aggregation occurs over all possible spatial coordinates i, j), and then the discrepancy between the feature distributions between stylization and style is penalized. Such a structure of style losses approximates the style, but does not lead to the transfer of content (semantic) information from the style image.

3. Proposed Modification of Style Transfer

Since being tied to a specific custom style image can be too restrictive, leading to styling artifacts, as shown in fig. 1, a three-stage styling algorithm is proposed that is more robust to individual incompatibilities between the original content and style images:

1. Find for the source style image the most similar images in style in a wide database of art images, such as Wikiart [9] or Pandora [10] (which was used in the work). Optional: recolor similar style images to the style's color scheme.
2. Average Gram matrices over similar images.
3. Apply the styling algorithm 1, replacing the style image's Gram matrices with averaged Gram matrices over a set of images that define a style similar to the given one.

To search for similar images by style, vectors of per-channel means for intermediate representations in the VGG network are extracted. For example, for the image Z and its intermediate representation Z^k on the layer k, we obtain the components of the C_k-dimensional mean vector as follows:

$$v_c^k(Z) = \frac{1}{H_k \cdot W_k} \sum_{i=1}^{W_k} \sum_{j=1}^{H_k} Z_{cij}^k, \; c = \overline{1, C_k}. \tag{4}$$

Further, these vectors are concatenated along the earlier and later VGG layers. Comparison of images by style is performed by comparing the resulting vectors according to the Euclidean norm. The specified vector representation contains information about the style, and not about the content, since channel-by-channel averaging erases information about the spatial arrangement of objects, and only the statistics of the presence of certain features (colors and more general local patterns) that characterize the style are saved.

Table 1
User survey results: comparing original [4] and proposed style transfer method.

num. of questions	num. of respondents	num. of answers	% of votes for the new method
25	58	1450	71%

To improve the correspondence with the original style, it is recommended to recolor the most similar style images to the original style in the colors of the original style, using the histogram matching algorithm [11].

Styling with only the input style image can create individual inconsistencies between it and the content image, causing styling artifacts. To reduce the risk of inconsistency in the approach proposed above, it is proposed to style using the usual method (1), but in style loss (2) to approximate the styling Gram matrix to the average Gram matrix of several images similar in style instead of the single target style image. This is analogous to using an ensemble of predictive models instead of a single model, which is often used to improve forecasting accuracy.

Although the next section shows the results of runs for the styling model [4], the proposed approach is general enough to be applicable to other styling methods. The code for the proposed approach is available at `https://github.com/valerapon/Style-transfer-UESC`.

4. Comparative Experiments

Here we define the parameters at which styling will be performed. The maximum number of similar style images is 3, we use Euclidean metric similarity metric, and the threshold of image proximity is set to $threshold = 500$. Styling will be carried out according to the aggregated (using averaging of Gram matrices) style.

Figures 3,4 show some examples of styling with and without recoloring. In both cases, you can see that the resulting stylization is slightly different from the original, although it repeats its general features. The lack of recoloring reduces the consistency with the original style, but can add brightness to the result, which is especially noticeable in fig. 3.

A survey was also conducted among respondents who, for all kinds of content-style pairs, were offered a choice of two styling options — the basic [4] method and the proposed one (with recoloring). Respondents were asked to choose, in their opinion, a more successful stylization option. Styling options were offered randomly each time, and the respondents were not familiar with the details of styling algorithms. A total of 58 respondents participated, each comparing 25 pairs of stylizations. Poll results are presented in the table 1.

The user survey result showed that users chose the new styling method in the majority of cases (71%), showing its superiority over the base [4] style transfer method.

5. Conclusion

In the work, the problem of automatic stylization of images was studied. It was proposed to move away from the original style image, replacing it with a group of similar images in style. For style

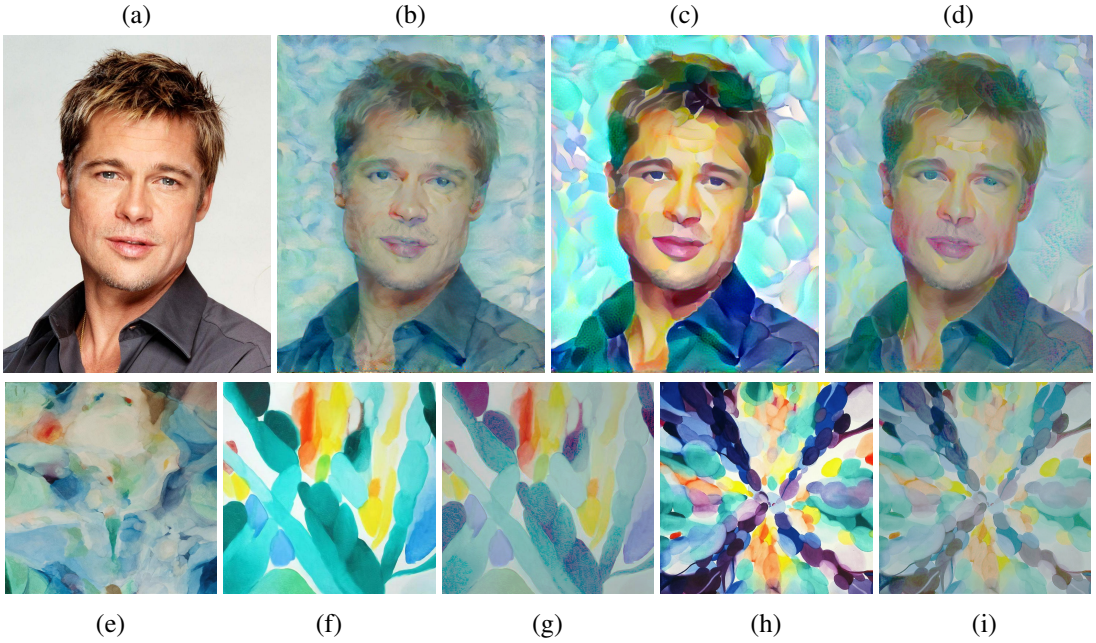


Figure 3: An example of the work of the styling algorithm for a group of found styles with recoloring the original style. (a) content; (b) styling (a) by (e); (c) styling by a group of found styles (f) and (h); (d) styling by style group with recoloring (g) and (i); (e) original style; (f) 1st style worn; (g) style (f) recolored in (e); (h) 2nd style found; (i) style (h) recolored in (e).

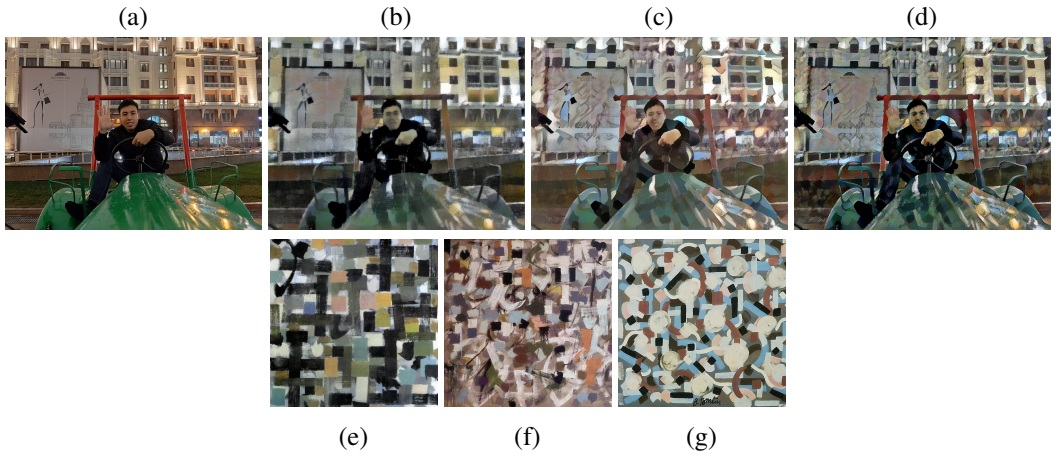


Figure 4: An example of the work of the styling algorithm on a group of found styles with the transfer of the color scheme of the original style. (a) – content; (b) – normal styling [4] (a) to (e); (c) – styling by a group of found styles (f) – (g); (d) – styling by style group with recoloring (f) – (g); (e) – user style; (f) – (g) styles similar to (e).

matching, the vectors of per-channel means in the VGG representation were used, which showed themselves well in practice. Styling with a group of similar styles, and not just with the original ones, gives a more stable result, which was shown by surveys of respondents who more often prefer stylizations by the proposed method. In addition, by changing the subset of similar style images used in style transfer, yield different stylization results, which is useful if the user is not satisfied with the initial stylization.

6. Acknowledgements

This research was performed in the framework of the state task in the field of scientific activity of the Ministry of Science and Higher Education of the Russian Federation, project "Models, methods, and algorithms of artificial intelligence in the problems of economics for the analysis and style transfer of multidimensional datasets, time series forecasting, and recommendation systems design", grant no. FSSW-2023-0004.

Bibliography

[1] B. Gooch, A. Gooch, Non-photorealistic rendering, AK Peters/CRC Press, 2001.
[2] T. Strothotte, S. Schlechtweg, Non-photorealistic computer graphics: modeling, rendering, and animation, Morgan Kaufmann, 2002.
[3] P. Rosin, J. Collomosse, Image and video-based artistic stylisation, volume 42, Springer Science & Business Media, 2012.
[4] L. A. Gatys, A. S. Ecker, M. Bethge, A neural algorithm of artistic style, CoRR abs/1508.06576 (2015). URL: http://arxiv.org/abs/1508.06576. arXiv:1508.06576.
[5] X. Huang, 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.
[6] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, M.-H. Yang, Universal style transfer via feature transforms, in: Advances in neural information processing systems, 2017, pp. 386–396.
[7] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).
[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.
[9] F. Phillips, B. Mackintosh, Wiki Art Gallery, Inc.: A Case for Critical Thinking, Issues in Accounting Education 26 (2011) 593–608. doi:10.2308/iace-50038.
[10] C. Florea, R. Condorovici, C. Vertan, R. Butnaru, L. Florea, R. Vrânceanu, Pandora: Description of a painting database for art movement recognition with baselines and perspectives, in: 2016 24th European Signal Processing Conference (EUSIPCO), 2016, pp. 918–922. doi:10.1109/EUSIPCO.2016.7760382.
[11] V. Buzuloiu, M. Ciuc, R. Rangayyan, C. Vertan, Adaptive-neighborhood histogram equalization of color images, J. Electronic Imaging 10 (2001) 445–459.