Artistic Style Discovery with Independent Components

Xin Xie[1], Yi Li[2,*], Huaibo Huang[3], Haiyan Fu[1], Wanwan Wang[4], Yanqing Guo[1]
[1]School of Information and Communication Engineering, Dalian University of Technology, China
[2]School of Artificial Intelligence, Dalian University of Technology, China
[3]Center for Research on Intelligent Perception and Computing, CASIA
[4]InsightOne Tech Co., Ltd., Beijing, 100028, China
shelsin@mail.dlut.edu.cn, {liyi, fuhy, guoyq}@dlut.edu.cn
huaibo.huang@cripac.ia.ac.cn, wangwanwan@insightone.cn

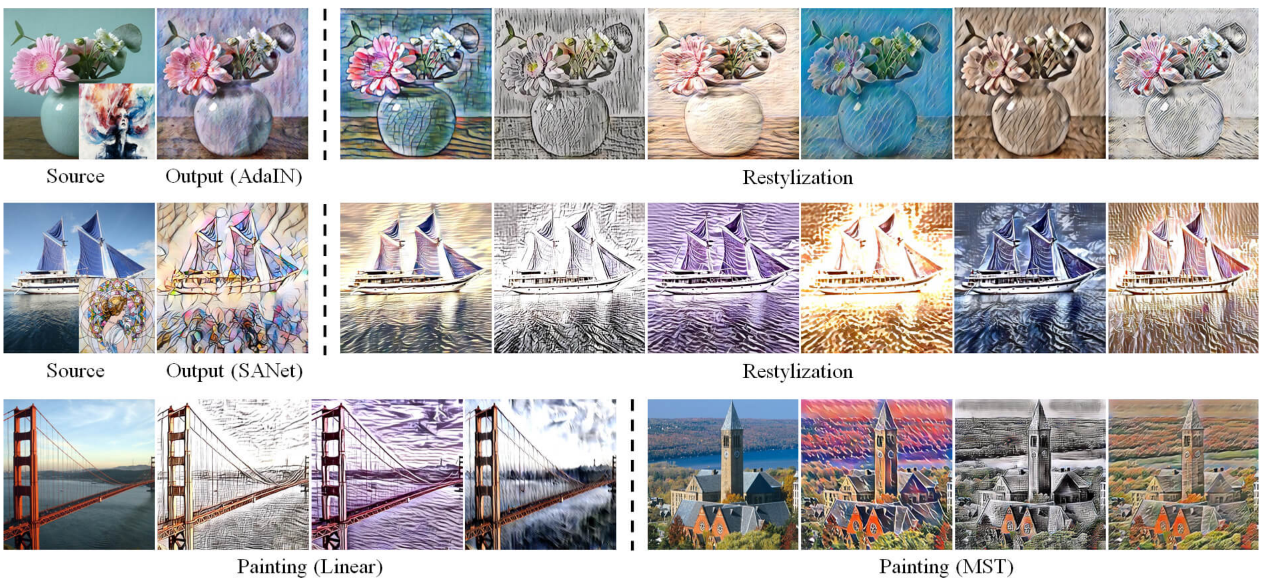

Figure 1. **Diverse restylized artworks from different backbones** including AdaIN [14], Linear [20], SANet [26] and MST [41]. In the first two rows, the first column is the source of the content image with the style image and the second column is the original artistic output, the other columns are the output images with artistic styles discovered by our algorithm. In the last row, given a natural scene, our method yields the other paintings.

Abstract

Style transfer has been well studied in recent years with excellent performance processed. While existing methods usually choose CNNs as the powerful tool to accomplish superb stylization, less attention was paid to the latent style space. Rare exploration of underlying dimensions results in the poor style controllability and the limited practical application. In this work, we rethink the internal meaning of style features, further proposing a novel unsupervised algorithm for style discovery and achieving personalized manipulation. In particular, we take a closer look into the mechanism of style transfer and obtain different artistic style components from the latent space consisting of different style features. Then fresh styles can be generated by linear combination according to various style components. Experimental results have shown that our approach is superb in 1) restylizing the original output with the diverse artistic styles discovered from the latent space while keeping the content unchanged, and 2) being generic and compatible for various style transfer methods. Our code is available in this page: https://github.com/Shelsin/ArtIns.

*Yi Li is the corresponding author.

1. Introduction

The target of artistic style transfer is to transform the style of one image into an arbitrary image, keeping the latter's content unchanged. The work in this topic has seen rapid development in both academy and industry recently. Convolutional neural networks (CNNs) [3, 8, 16] is frequently used to achieve stylization because of its superior learning property and excellent performance, especially VGG-net [35]. Besides CNNs, style transfer methods also require large-scale datasets and more advanced neural networks to improve the quality of image generation. Nevertheless, it is still challenging to produce satisfying and convincing artworks.

Starting from neural style transfer (NST) [7], recent works tend to generate high-quality stylized image via two main ways: iterative optimization [6, 23, 29] and feed-forward network [9, 16, 40]. With the former, the prior works rely on an optimization process to flexibly combine content and style of arbitrary images, which requires a long time to achieve stylization. The later achieves fast stylization with a single forward pass, but a major limitation of most feed-forward methods is that each network is restricted to a fixed number of styles.

While achieving impressive results in artistic style transfer, existing methods focus much on style extraction and content reconstruction but ignore the significance of the latent space. Although underlying knowledge learned by these prior works can be transformed into the final output, the real sense of them is less explored. In other words, the essence of the stylization process is projecting the original style features from the image space into the intermediate space, then the decoder transforms the latent knowledge back into the image space. During the process of style transfer, most existing methods ignore the rich latent information contained in the image space, which conducts limited style controllability and diversity. Meanwhile, most of the previous methods need advanced devices like GPU with large memory to train their models for diverse styles, which limits their application to real-world scenarios.

In order to solve the problem, we propose a generally applicable approach to discover a large number of artistic styles via mathematical computation, which is unsupervised and independent of any form of training. Our method is named ArtIns as the short for *Artistic Ingredients/Components Separation*. In this work, we take a step back and rethink the whole transfer process. After examining the relation between latent style features and the image variation, we find that versatile styles can be divided into multiple independent style components. According to these style components, fresh styles can be obtained to produce high-quality and controllable stylization results. In other words, diverse artistic styles are discovered from the latent space consisting of various style representations, which is exhibited in Fig. 1. Extensive experiments based on popular pre-trained style transfer models(*e.g.*, AdaIN [14], Linear [20], SANet [26] and MST [41]) indicate the effectiveness and flexibility of our algorithm.

To summarize, the main contributions are as follows:

- We introduce a novel unsupervised algorithm that can discover various styles from the latent space, advancing the ability of controllable stylization.
- We obtain the independent style components from the mixed latent style dimensions in style transfer, resulting in multiple artistic stylizations and lowering computational costs.
- Our method is generally applicable without training and we demonstrate the effectiveness and flexibility of our approach via abundant experiments on several state-of-the-art style transfer models.

2. Related Works

2.1. Style Transfer

Style transfer aims to migrate the style of a painting to a photograph, maintaining its original content. Initially, Gatys et al [7] proposed an iterative method to transfer styles by jointly minimising the distance of a white noise image from the content representations and the style representations. An increasing number of methods have been developed thereafter to advance the quality of stylization. For instance, AdaIN [14], normalizing the mean and variance of each feature map separately, was proposed to adaptively combine the content and style. Considering the limitation of learning a single style, transferring multiple styles into one image with common content [3] is achieved by introducing conditional instance normalization. Besides, Avatar-Net [33] utilizes a framework like U-net [30] to make up the content features by semantically aligned style features, preserving detailed style patterns and generating more plausible artworks. Arbitrary style transfer in real-time [16] is improved by minimizing perceptual loss which is the combination of feature reconstruction loss as well as the style reconstruction loss. Different from these, we set about style discovery from the aspect of the latent space and achieve various restylized results with low computation costs.

2.2. Image Editing

Disentangled representation is an unsupervised learning technique that breaks down each feature into narrowly defined variables and encodes them as separate dimensions. Actually, many methods of unsupervised disentanglement [2, 13, 17] have been achieved in general Generative Adversarial Networks (GANs) [4, 5, 10, 12] and Variational auto-encoders (VAEs) [18], since the former is an unsupervised training method via unlabeled datasets and the latter can achieve similar effects by building a self-reconstruction loss. Prior works [31, 39, 42] have already trained a linear classifier to accomplish the disentanglement of latent dimensions for image editing, making remarkable results. An unsupervised method of orthogonalization [32] is introduced to discover latent semantics to generate realistic images with target attributes. The latent space distiller [34] is used to achieve attribute disentanglement and GANSpace [11] performs PCA on the sampled data to find primary directions in the latent space. However, most of these methods only work under certain conditions, such as a clearly defined target attribute classifier, data sampling or model training. It is challenging and tough to separate latent style directions since the relation among underlying dimensions is complicated, especially in style transfer. In this paper, we propose an algorithm to discover various artistic styles via algebraic operation without any parameters and data to learn or train.

3. Method

We introduce an unsupervised algorithm ArtIns to discover diverse styles from the latent space consisting of diverse style features. Specifically, we rethink the sense of the style features and find that the latent style representations may be composed of multiple independent style components. These style components can be captured from the latent style space by mathematical operations. Finally, new styles are synthesized by linearly combining style ingredients with different coefficients.

3.1. Basic Architecture of Style Transfer Models

In style transfer, the basic architecture of most existing methods [7, 22, 23, 33] is the encoder-decoder network framework. Both the content image of photograph and the style images of painting $I_c, I_s \in R^{h\times w\times c}$ are first transformed into different latent space $C, S \subseteq R^n$ via a encoder network E. Then content features $c \in C$ and style features $s \in S$ are projected into the common n-dimensional latent space $Z \subseteq R^n$ by an affine transformation. After that, decoder D is responsible for generating novel image $I \in R^{h\times w\times c}$ with latent code $z \in Z$ combining content features and style features. Generally speaking, advanced encoder plays an essential role in the whole style transfer process for more detailed and accurate style features, which can help the decoder produce high-quality stylized artwork. A pre-trained VGG-net [35] is usually selected as the encoder to obtain content representations and style representations. For better discussion, we first denote the encoding process of a style image as follow:

$$s = E(I_s) \tag{1}$$

3.2. Rethinking the Role of Style Features s

In order to study how the stylized image behaves when the style representations change, we take a deep look to the whole process of style transfer. Taking AdaIN model [14] as an example for analysis, the latent code z is projected to a stylized image layer by layer via the decoder. The image variation may be related to the internal ingredients of the latent code z. Considering the style representations, we set μ_s and σ_s denotes the mean and standard deviation of the style image features s respectively. AdaIN [14] applies a style-defined affine transformation to normalize the content features c shown as Eq. (2):

$$z = AdaIN(c, s) = \sigma_s(\frac{c - \mu_c}{\sigma_c}) + \mu_s \tag{2}$$

where μ_c and σ_c are the mean and standard deviation over the content feature channel. We name the first convolution layer of the decoder network as $conv1$, which directly acts on the latent code z. Based on Eq. (2), the partial convolution effect $conv1(s)$ can be separated from $conv1(z)$ because of μ_s and σ_s. So we suppose that the latent style feature s is the root factor to control the whole style of final generated artwork. However, we gain no rule of the change from one image style to another, resulting in the poor style controllability.

3.3. Exploration of Internal Style Space

In recent decades, there are numerous applications where the data can be represented by the linear combination of other samples. For instance, each individual face can be represented exactly as the linear combination of eigenfaces [1]. Inspired by the Fourier transform [38], time-domain and frequency-domain analysis of signals [25, 27, 36], we argue that the style features s is mixed after encoding the style image, which can be seen as the discrete series shown in Eq. (3):

$$s_i = [sty_{i1}, sty_{i2}, \cdots\cdots, sty_{in}] \tag{3}$$

where $s_i \in S$ denotes the latent style representations of the i-th style image. And sty_{it} represents the t-th dimensional representation of s_i, which can be regarded as the sample value in the discrete series at the time $t = 1, 2, 3, \cdots, n$. According to the discrete Fourier transform [37], an arbitrary signal can be easily manipulated merely by representing it as a linear combination of simple and mathematically well-defined signals. In other words, the style features s_i can be separated as multiple independent components. Linear combination of these independent components with different coefficients results in versatile styles.

3.4. Artistic Ingredients Separation

In Sec. 3.2 and Sec. 3.3, we take a step back and reanalyze the sense of the style features, which consists of

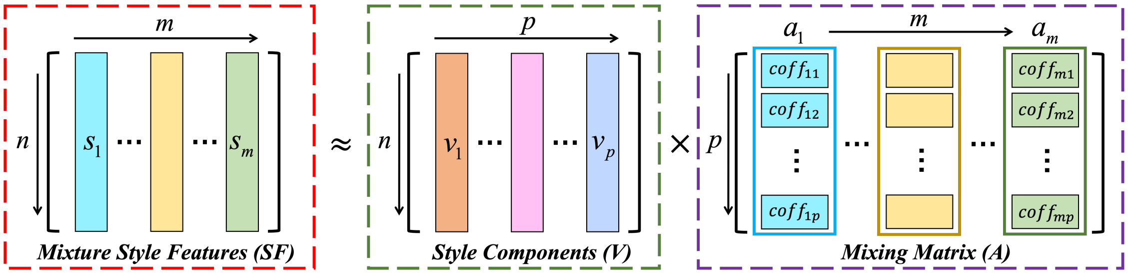

Figure 2. **The style features are linear sum of style components** where the mixed matrix $SF_{n\times m}$ can be divided into the mixing matrix $A_{p\times m}$ and style components $V_{n\times p}$.

various independent style components. We define one single style component vector $v_x \in R^n$ as Eq. (4) where $x = 1, 2, 3, \cdots, p$ and p represents the number of independent style component vectors.

$$v_x = [ele_{x1}, ele_{x2}, \cdots\cdots, ele_{xn}] \tag{4}$$

Here, ele_{xn} represents the n-th element in the component vector v_x. For the i-th style image, the latent style feature vector s_i can be denoted as the linear sum of p style component vectors, which is shown in Eq. (5):

$$s_i = V \cdot a_i = v_1 \cdot coff_{i1} + v_2 \cdot coff_{i2} + \cdots + v_p \cdot coff_{ip} \tag{5}$$

where $a_i = [coff_{i1}, coff_{i1}, \cdots, coff_{ip}]$ is the coefficient of linear sum for independent component vectors. We sample m style images and utilize the style encoder to transform them into latent style representations, which are ingredients of the style feature matrix $SF \in R^{n\times m}$. The coefficients of linear sum for each style vector can be constructed as the coefficient matrix $A \in R^{p\times m}$ and p independent style component vectors is defined as matrix $V \in R^{n\times p}$. Then we can gain the relation as follow:

$$SF_{n\times m} = V_{n\times p} \cdot A_{p\times m} \tag{6}$$

By observing Eq. (6), we suppose that it can be transformed into the cocktail party problem. In details, matrix V is the source signals and the coefficient matrix A is a mixing matrix. Then the style features matrix SF can be seen as the mixture signals. The whole mixing steps is shown in Fig. 2. In this work, we make advantages of FastICA [15, 19] algorithm to obtain each independent style component vector.

3.5. Implementation on Artistic Models

We have introduced an unsupervised algorithm to discover styles from latent space consisting of diverse style features. Actually, our algorithm can be effectively performed in various style transfer methods which are based on the encoder-decoder network framework. In this part, we describe how to embed our algorithm into representative style transfer networks, such as **AdaIN** [14], **Linear** [20], **SANet** [26] and **MST** [41].

All of the artistic methods above conduct a combination of content features and style features in different ways. For instance, AdaIN [14] has no learnable affine parameters to integrate two kinds of features by simply aligning the channel-wise mean and variance of content features to match those of style features. Linear [20] and SANet [26] achieve the feature fusion via a module that contains several convolutional transformations. After centering two kinds of features by subtracting their mean vectors, MST [41] blends content features with transferred features made by feature whitening and coloring as used in WCT [21]. No matter how these methods integrate content code and style code, we obtain independent style components only from multiple style features after encoding. We gather these latent style representations like a combination and style components can be separated from the latent combinative space. Finally, changing the style code according to each style component vector to generate fresh stylized images.

3.6. Discussion

Difference from Interpolation. Both the method of interpolation and our algorithm can achieve the diversity of fresh stylized images, but the computational cost and the stylized effect make difference. Compared to the method of interpolation, our algorithm only requires a small dataset of style images to separate basic style ingredients. These style components can be reconstructed back into the original styles or even be linearly grouped into various new styles that do not exist in the style dataset. For the interpolation in artistic models, a large-scale style dataset is needed to bulid multiple style features via the encoder instead of algebraic manipulation, which is time and energy consuming. And the generated artwork is equipped with the mixture of different style features because of interpolation, which is difficult to control the changing direction of artistic styles.

Deep Network vs. Conventional Signal Analysis. At start, we have considered to use a deep network to achieve the decomposition but find it inappropriate. 1) We need

Figure 3. **Various generated artworks via random linear combination of style components** conducted in four methods including AdaIN [14] (the first two rows), Linear [20] (the third row), SANet [26] (the fourth and fifth rows) and MST [41] (the last row). The first column is the source image of the content with the source image of the style and the second column is the original output. The other columns are diverse artworks with new styles generated by setting random coefficients of linear sum.

to set an appropriate loss function between every two style components to decrease their correlation, which is a super large computation cost. 2) The essence of feature maps is the mixture of the most style components, which is hard to learn the independent style component. 3) It is inconvenient to retrain the network when embedded into different artistic methods.On the other hand, it is more reasonable to explore the style feature than the raw image. We argue that dividing the images only obtain the external pixel information about RGB channels, while the style feature holds compact and abstract style information. ArtIns can be applied to the generated artworks and reconstruct the original output.

4. Experiments

In this section, we conduct different experiments on a wide range of artistic models, which indicates the flexibility of our method. We first introduce the basic preparation and related parameters setting before processing experiments. Then we evaluate the performance of our method, using our algorithm to discover various styles. Besides, we utilize detached style components to achieve image editing and beautification, which demonstrates the application scenarios of our method. In a nutshell, we discuss the following topics in this section:

- Whether our algorithm can generate diverse artistic styles?
- Does these independent style components make sense?
- Whether the original styles can be reconstructed via independent components?

4.1. Artistic Models and Datasets

Artistic Models. We evaluate the unsupervised algorithm on four state-of-the-art artistic models, AdaIN [14],

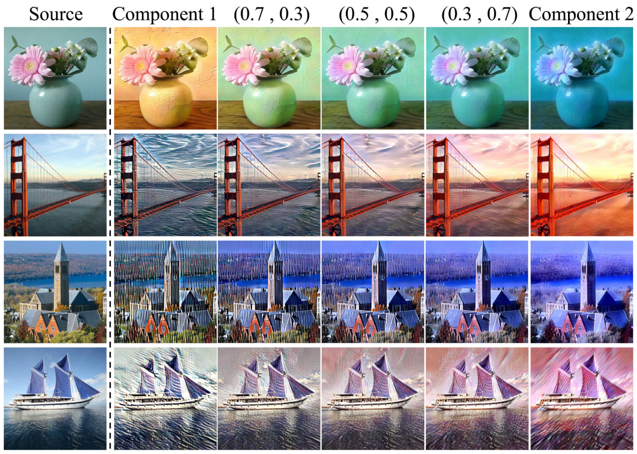

Figure 4. **Interpolation of style components** on SANet [26]. The first column is the image of natural scenes and the other columns are the variation between two artistic style components discovered by our method. The parameters in parentheses represent the proportion of two style components.

Linear [20], SANet [26] and MST [41] respectively. Our experiments of these methods can be simply operated on both CPU and general GPU devices.

Datasets. We adopt the MS-COCO dataset [24] as the content images and the WikiArt dataset [28] as the style images. Both datasets contain tens of thousands of images and we resize every input image to 512×512 in our experiment. Based on a large body of style data, we demonstrate that a sufficient number of examples and style components is required for the ICA decomposition. Unless otherwise specified, we sample 512 style images from the WikiArt [28] to be projected into the latent style space and we assume each style vector consists of 512 independent style component vectors.

4.2. The Diversity of Generated Artistic Styles

In Sec. 3, we make a deep analysis of latent style features and discover that each style vector can be decomposed into multiple independent style component vectors. These style ingredients can be linearly grouped into another fresh style which do not exist in the style dataset. We verify the diversity of generated styles in two ways: one is making a linear combination of style components with random coefficients, the other is conducting the interpolation of style components.

Linear Combination of Style Components. In order to verify the effectiveness of our method, we set random coefficients of linear sum to build diverse new styles as shown in Fig. 3. The experimental results are obtained on four artistic models, which are AdaIN [14], Linear [20], SANet [26] and MST [41] respectively. By observing the results, we can find that there is significant difference between the original stylized image and other generated artworks with random styles. Each fresh artwork not only owns its special style features but also keeps its original content features well. We utilize different coefficients to achieve linear sum of style components, resulting in various new styles, which demonstrates the diversity of artistic styles discovered by our method. Meanwhile, the results tell us that our algorithm possesses strong flexibility.

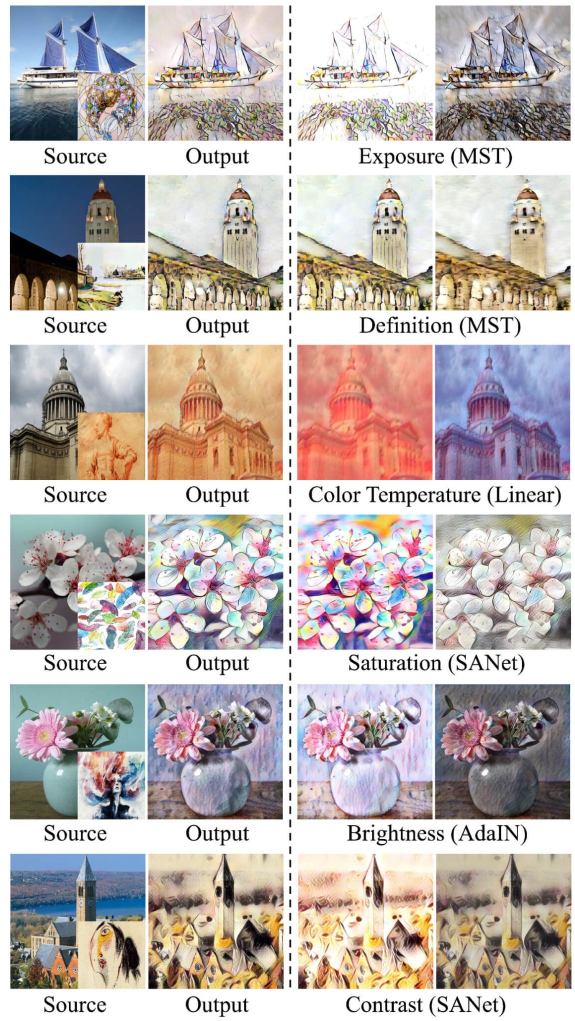

Figure 5. **Artwork adjustment based on color tone in different artistic methods**, including AdaIN [14], Linear [20], SANet [26] and MST [41]. For each set of images, the first column is the source of the content image with the style image, the second column is the original artistic output, and the last two columns are the output images by changing the style code according to style components found by our method.

Interpolation of Style Components. We conduct our experiments on SANet [26] to demonstrate the diversity of artistic styles. Fig. 4 gives some experimental results. From the results, we gain that each component vector has its spe-

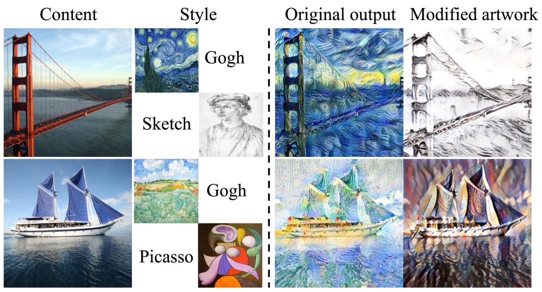

Figure 6. **Artwork adjustment** on SANet [26]. We give two style images from Van Gogh, then we adjust the coefficients to make the original artworks become the sketch and Picasso styles which are not in the datasets.

cial style features and controls different style effects. Different style components can be interpolated to get diverse styles. For example, we can discover two components that control the yellow and blue respectively as shown in the first row in Fig. 4. By setting different interpolation parameters, we can obtain a change in style from yellow to green to blue. Similarly, we can overlay other style components controlling the line, texture, shape, etc, which also verifies the latent style representations can be divided into multiple independent components.

4.3. The Sense of Style Components

In this part, we show more effective experimental results to verify the meaning of independent style components. Next step, we study how to integrate these style components to achieve the image editing and beautification.

Color Tone for Artwork. By abundances of experiments, we discover that some style components can control different variation. These components can be given explicit property definitions, such as exposure, brightness, definition, contrast, saturation, color temperature, etc. Fig. 5 gives some examples. In order to modify related categories, we allow four methods, including AdaIN [14], Linear [20], SANet [26] and MST [41], to extend their latent style code by increasing the proportion of corresponding artistic style components discovered by our algorithm. In this way, we can edit and beautify various artworks. In some situations, it may be more convenient to find a suitable style image. However, the available artworks are limited in practice and may not convey the desired style. When there is no proper reference by hand, our method provides effective solution by modifying the components. Besides primary characteristics as brightness and contrast, ArtIns can also edit more complex attributes, including the textures of wave, sketch and spot. Fig. 6 gives an example.

Stylization of Natural Scenes. Generally speaking, the image of the natural scene needs the image of paintings as the style guideline for most artistic models, which is time and energy consuming to collect a large number of style images and increase the computational costs. In this work, we find that some special style components can control different artistic effects, such as sketch, oil painting, etc. For the natural image, we use the encoder to make its style as the original style features. Then we can replace or extend the latent style representation to stylize the natural scene without other style images as guideline. Fig. 7 shows experimental results based on four models, including AdaIN [14], Linear [20], SANet [26] and MST [41].

Figure 7. **Versatile artworks of natural scene stylization** on AdaIN [14], Linear [20], SANet [26] and MST [41]. For each set of images, the first column is the image of natural scenes and the other columns are the fresh artistic outputs with new styles discovered by our method.

4.4. The Effect of Style Reconstruction

In order to verify the original styles can be recovered with independent style components, we utilize the mixing matrix A and style components V to reconstruct the original styles as much as possible. Fig. 8 shows the results of style reconstruction based on four artistic models, including AdaIN [14], Linear [20], SANet [26] and MST [41]. By observing the results, we argue that the style features can be exactly divided into a lot of independent style components and these components can be linearly combined into the original styles. Besides, we compute the mean square error (MSE-loss) between the style code of style image and that of the original artistic output (OS). We also compute the style loss of style reconstruction (RS). Tab. 1 gives the specific information. We argue that it is challenging to completely recover the original styles because these style components are not absolutely independent as discussed in Sec. 4.5, which requires further study.

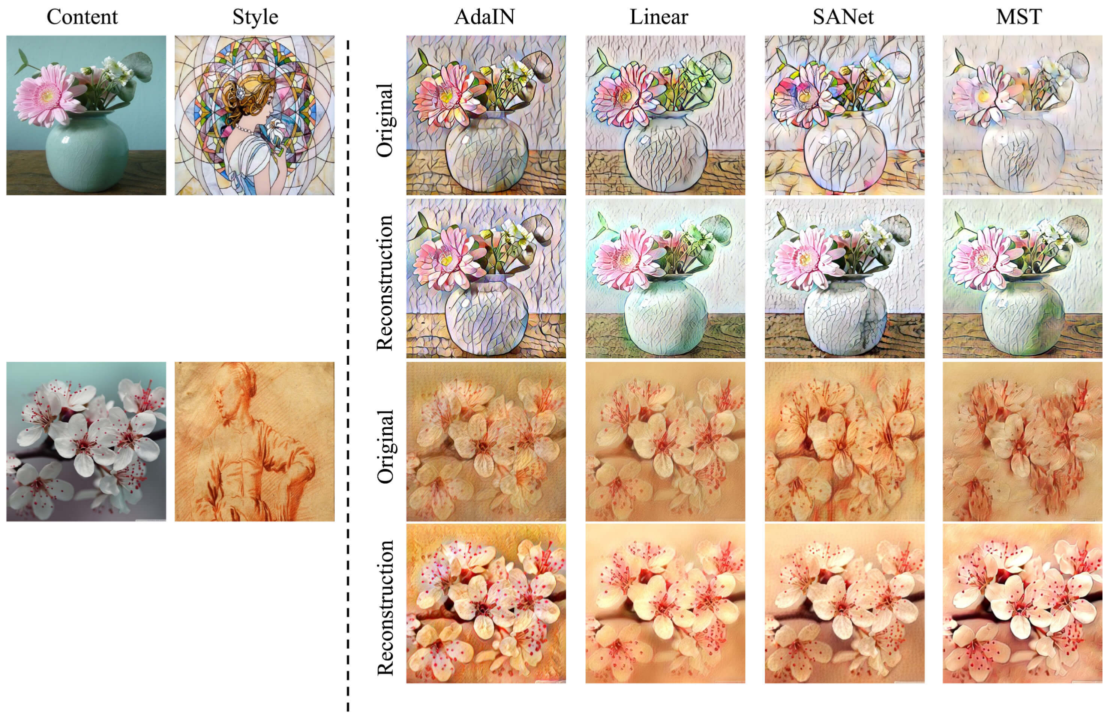

Figure 8. **Style Reconstruction** on AdaIN [14], Linear [20], SANet [26] and MST [41] according to the mixing matrix A. For each set of images, the first two columns are the image of content and style respectively. The other columns of the first row are the original artistic outputs and that of the second row are the fresh artworks with reorganized styles.

Table 1. The style loss of the output before and after style reconstruction.

Method	OS	RS
AdaIN [14]	0.29	1.17
Linear [20]	0.09	0.34
SANet [26]	0.04	0.19
MST [41]	0.25	0.69

4.5. Limitations

Our method is achieved under the assumption that the style features are linearly combined by multiple independent style components. Hence ArtIns may not be well applied if the style features are nonlinearly combined by style components, which is hard to be defined for its complex ingredient composition. In our experiments, we find that there exists style components that effect more than one element. For example, the line width would also be changed when we modify a color. But we also discover that the component of *green* can be replaced by linearly combing the components of *yellow* and *blue*. There might be some nonlinear relation between style components, which is rather challenging and complex to learn.

5. Conclusion

In this paper, we have proposed a novel unsupervised algorithm to address the style controllability by discovering artistic styles with independent components. Specifically, we rethink the meaning of the style features and find that the latent style representations can be divided into multiple style ingredients. We take full advantage of rich latent knowledge to obtain these independent style components. Then various fresh styles can be generated by linearly assembling the style components. Extensive experiments on four artistic methods show that our algorithm not only controls the style changing direction of the images but also is generic and compatible for various style transfer methods. Compared to state-of-the-art methods, our unsupervised method requires no parameter to learn or train, which is more flexible, efficient and practical.

Acknowledgement: This work is supported by the National Natural Science Foundation of China (No. 62106037, No. 62076052, No. U1936117), the Fundamental Research Funds for the Central Universities (DUT20RC(3)088, DUT20TD110), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (No. 202100032) and CCF-Baidu Open Fund (No. 2021PP15002000).

References

[1] Mayank Agarwal, Nikunj Jain, Mr Manish Kumar, and Himanshu Agrawal. Face recognition using eigen faces and artificial neural network. *International Journal of Computer Theory and Engineering*, 2(4):624, 2010. 3

[2] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016. 2

[3] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016. 2

[4] Chaoyou Fu, Yibo Hu, Xiang Wu, Guoli Wang, Qian Zhang, and Ran He. High-fidelity face manipulation with extreme poses and expressions. *IEEE Transactions on Information Forensics and Security*, 16:2218–2231, 2021. 2

[5] Chaoyou Fu, Xiang Wu, Yibo Hu, Huaibo Huang, and Ran He. Dvg-face: Dual variational generation for heterogeneous face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[6] Leon A Gatys, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Preserving color in neural artistic style transfer. *arXiv preprint arXiv:1606.05897*, 2016. 2

[7] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 2, 3

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2

[9] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*, 2017. 2

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3:2672–2680, 2014. 2

[11] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020. 3

[12] Ran He, Jie Cao, Lingxiao Song, Zhenan Sun, and Tieniu Tan. Adversarial cross-spectral face completion for nir-vis face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 42(5):1025–1037, 2019. 2

[13] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016. 2

[14] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[15] Aapo Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999. 4

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2

[17] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 2

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[19] Zbynek Koldovsky, Petr Tichavsky, and Erkki Oja. Efficient variant of algorithm fastica for independent component analysis attaining the cramér-rao lower bound. *IEEE Transactions on neural networks*, 17(5):1265–1277, 2006. 4

[20] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3809–3817, 2019. 1, 2, 4, 5, 6, 7, 8

[21] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *arXiv preprint arXiv:1705.08086*, 2017. 4

[22] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018. 3

[23] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017. 2, 3

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[25] Dimitris G Manolakis and Vinay K Ingle. *Applied digital signal processing: theory and practice*. Cambridge university press, 2011. 3

[26] Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5880–5888, 2019. 1, 2, 4, 5, 6, 7, 8

[27] Abraham Peled and Bede Liu. Digital signal processing: theory, design, and implementation. *New York*, 1976. 3

[28] Fred Phillips and Brandy Mackintosh. Wiki art gallery, inc.: A case for critical thinking. *Issues in Accounting Education*, 26(3):593–608, 2011. 6

[29] Eric Risser, Pierre Wilmot, and Connelly Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017. 2

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[31] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2020. 3

[32] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1532–1540, 2021. 3

[33] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8242–8250, 2018. 2, 3

[34] Mustafa Shukor, Xu Yao, Bharath Bhushan Damodaran, and Pierre Hellier. Semantic and geometric unfolding of stylegan latent space. *arXiv preprint arXiv:2107.04481*, 2021. 3

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2, 3

[36] Petre Stoica, Randolph L Moses, et al. Spectral analysis of signals. 2005. 3

[37] Duraisamy Sundararajan. *The discrete Fourier transform: theory, algorithms and applications*. World Scientific, 2001. 3

[38] G Wakefield. Fourier series and the discrete fourier transform: Quick primer. *University of Michigan, EECE 206F01.*, 2001. 3

[39] Guoxing Yang, Nanyi Fei, Mingyu Ding, Guangzhen Liu, Zhiwu Lu, and Tao Xiang. L2m-gan: Learning to manipulate latent space semantics for facial attribute editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2951–2960, 2021. 3

[40] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2

[41] Yulun Zhang, Chen Fang, Yilin Wang, Zhaowen Wang, Zhe Lin, Yun Fu, and Jimei Yang. Multimodal style transfer via graph cuts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5943–5951, 2019. 1, 2, 4, 5, 6, 7, 8

[42] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pages 592–608. Springer, 2020. 3