CCPL: Contrastive Coherence Preserving Loss for Versatile Style Transfer

Zijie Wu[1*], Zhen Zhu[2*], Junping Du[3], and Xiang Bai[1†]

[1] Huazhong University of Science and Technology, China
[2] University of Illinois at Urbana-Champaign, USA
[3] Beijing University of Posts and Telecommunications, China
{zijiewu,xbai}@hust.edu.cn, zhenzhu4@illinois.edu, junpingdu@126.com

Abstract. In this paper, we aim to devise a universally versatile style transfer method capable of performing artistic, photo-realistic, and video style transfer jointly, without seeing videos during training. Previous single-frame methods assume a strong constraint on the whole image to maintain temporal consistency, which could be violated in many cases. Instead, we make a mild and reasonable assumption that global inconsistency is dominated by local inconsistencies and devise a generic Contrastive Coherence Preserving Loss (CCPL) applied to local patches. CCPL can preserve the coherence of the content source during style transfer without degrading stylization. Moreover, it owns a neighbor-regulating mechanism, resulting in a vast reduction of local distortions and considerable visual quality improvement. Aside from its superior performance on versatile style transfer, it can be easily extended to other tasks, such as image-to-image translation. Besides, to better fuse content and style features, we propose Simple Covariance Transformation (SCT) to effectively align second-order statistics of the content feature with the style feature. Experiments demonstrate the effectiveness of the resulting model for versatile style transfer, when armed with CCPL.

Keywords: image style transfer, video style transfer, temporal consistency, contrastive learning, image-to-image translation.

1 Introduction

Over the past years, much progress has been made on style transfer to make the result exceptionally pleasant and artistically valuable. In this work, we are interested in *versatile style transfer*. Apart from artistic style transfer and photo-realistic style transfer, our derived method is versatile in performing video style transfer well without explicitly training with videos. The code is available at https://github.com/JarrentWu1031/CCPL.

One naive solution to produce a stylized video is to independently transfer the style of successive frames with the same style reference. Since no temporal consistency constraint is enforced, the generated video usually has obvious

∗ Equal contribution † Corresponding author

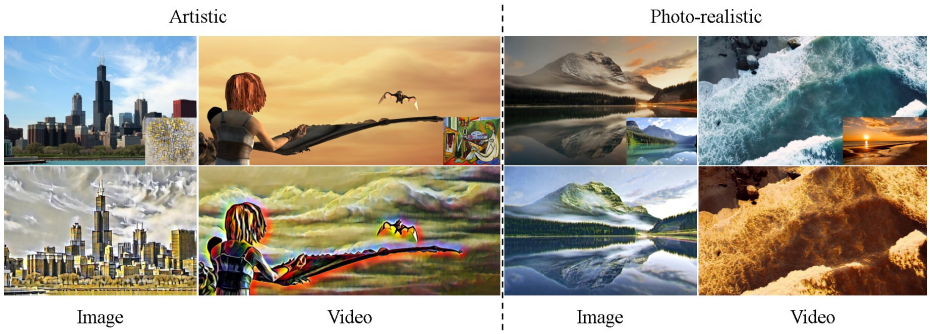


Fig. 1. Our algorithm can perform versatile style transfer. From left to right are examples of artistic image/video style transfer, photo-realistic image/video style transfer. Adobe Acrobat Reader is recommended to see the animations.

flicker artifacts and incoherence between two consecutive frames. To combat this problem, former methods [4,16,19,22,39,40] used optical flow as guidance to restore the estimated motions of the original videos. However, estimating optical flow requires much computation, and the accuracy of estimated motions tightly constrains the quality of the stylized video. Recently, some algorithms [14,29,33] tried to improve the temporal consistency of the video outputs with single-frame regularizations. They attempted to ensure a linear transformation from the content feature to the fused feature. The underlying idea is to encourage preserving the dense pairwise relations within the content source. However, without explicit guidance, the linearity is largely affected by the global style optimization. Therefore, their video results are still not that temporally consistent. We notice that most video results show good structure rigidity to their content video inputs, but the local noise escalates the impression of inconsistency. So instead of considering a global constraint that could be easily violated, we start by thinking about a more relaxed constraint defined on local patches.

As shown in Fig. 2, our idea is simple: the change between patches denoted by R'_A and R'_B of the same location in the stylized images should be similar to patches R_A and R_B of two adjacent content frames. If the two consecutive content frames are shot within a short period, it is likely to find a similar patch to R_B in the neighboring area, which is denoted by R_C (in the blue box). In other words, we can treat two nearby patches in the same image as patches of the same location in consecutive frames. Therefore, we can apply the constraint even when we only have single-frame images. However, forcing these patch differences to be the same is unreliable since it will encourage the outputs to be the same as the content images. Then no style transfer effects would appear in the results. Inspired by recent advances in contrastive learning [8,35,37], we use the InfoNCE loss [35] to maximize the mutual information between the positive pair (from the same region) of patch differences relative to other negative pairs (from different regions). By sampling a sufficient number of negative pairs, the loss encourages the positive pair to be close while keeping away from negative samples. We call the derived loss as **Contrastive Coherence Preserving Loss (CCPL)**.

After applying CCPL, we note that the temporal consistency of the video outputs improves substantially while the stylization remains satisfying (see Fig. 5

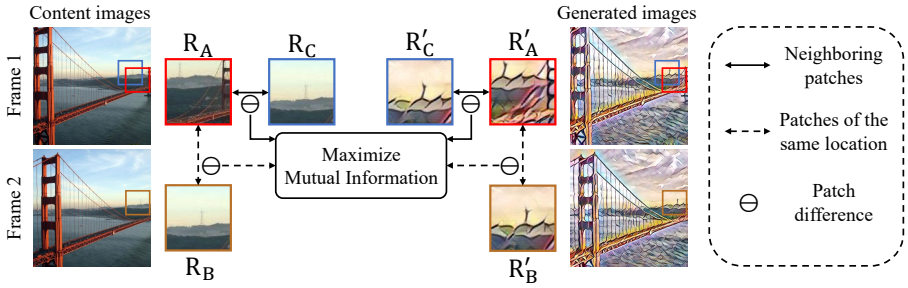


Fig. 2. Intuition of the contrastive coherence preserving loss. The regions denoted with red boxes from the first frame (R_A or R'_A) have the same location with corresponding patches in the second frame wrapped with brown box (R_B or R'_B). R_C and R'_C (in the blue boxes) are cropped from the first frames but their semantics align with R_B and R'_B. The difference between two patches is denoted as $\mathcal{D}$ (*e.g.*, $\mathcal{D}(R_A, R_B)$). The mutual information between $\mathcal{D}(R_A, R_C)$ and $\mathcal{D}(R'_A, R'_C)$ ($\mathcal{D}(R_A, R_B)$ and $\mathcal{D}(R'_A, R'_B)$) is encouraged to be maximized to preserve the coherence of the content source.

and Tab. 1). Besides, due to the neighbor-regulating strategy of the CCPL, the local patches of the generated image are constrained by their neighboring patches, which reduces local distortions significantly, thus leading to better visual quality. Our proposed CCPL does not require video inputs and is not bound to specific network architecture. Therefore we can apply it to any existing image style transfer networks during training to improve their performance on images and videos (see Fig. 9 and Tab. 1). The significant improvement in visual quality and its flexibility empowers CCPL for photo-realistic style transfer with minor modifications, marking it a vital tool towards versatile style transfer (see Fig. 1).

With CCPL, we now aspire to fuse content and style features both efficiently and effectively. To realize this, we propose an efficient network for versatile style transfer, called **SCTNet**. The critical element of SCTNet is the **Simple Covariance Transformation (SCT)** module to fuse style features and content features. It computes the covariance of the style feature and directly multiplies the feature covariance with the normalized content features. Compared to the fusing operations in AdaIN [23] and Linear [29], our SCT is simple and can capture precise style information at the same time.

To summarize, our contributions are three-fold:

1. We propose Contrastive Coherence Preserving Loss (CCPL) for versatile style transfer. It encourages consistency between the content image and generated image in terms of the difference of an image patch with its neighboring patches. It is effective and transferable to other style transfer methods.
2. We propose Simple Covariance Transformation (SCT) to align second-order statistics of content and style features effectively. The resulted SCTNet is structurally simple and remains efficient (about 25 frames per second at the scale of 512×512), which is of great potential for practical use.
3. We apply our CCPL to other tasks, such as image-to-image translation, and improve the temporal consistency and visual quality of results without further modifications, demonstrating the flexibility of CCPL.

2 Related Works

Image Style Transfer. These algorithms aim at generating an image with the structure of one image and the style of another. Gatys *et al*. first pioneered Neural Style Transfer (NST) [17]. For acceleration, some algorithms [25,45] approximated the iterative optimization procedure as feed-forward networks and achieved style transfer with a fast forward pass. For broader applications, several algorithms tried to transfer multiple styles within a single model [5,15]. Nevertheless, these models have limitations on the number of learnt styles. Since then, various methods have been designed to transfer style from random images.

Style-swap methods [7,42] swapped each content patch with its closest style patch before reconstructing the image. WCT [30] utilized singular value decomposition to whiten and then re-color images. AdaIN [23] replaced the feature means and standard deviations with those from the style source. Recently, many attention-based algorithms came forth. For example, Li *et al*. [29] devised a linear transformation to align second-order statistics between the fused feature and the style feature. Deng *et al*. [14] improved it with multi-channel correlating. SANet [36] re-arranged style features utilizing spatial correlations with content features. AdaAttN [33] combined AdaIN [23] and SANet [36] to balance global and local style effects. Cheng *et al*. [11] proposed style-aware normalized loss to balance stylization. Another branch aims to transfer photo-realistic style onto images. Luan *et al*. [34] designed a color transformation network inspired by the Matting Laplacian [28]. Li *et al*. [31] replaced the upsampling layers of WCT [30] with unpooling layers and added max-pooling masks to alleviate detail losses. Yoo *et al*. [47] introduced the wavelet transform to preserve structural information. An *et al*. [2] used neural architecture search algorithms to find the appropriate decoder design for better performance.

Video Style Transfer. Existing video style transfer algorithms can be roughly divided into two categories according to whether to use the optical flow or not.

One line of work leverages optical flow when producing the video output. These algorithms try to estimate the motion of the original video and restore it in the generated video. Ruder *et al*. [39] proposed a temporal loss to regulate the current frame with the warped previous frame to extend the image style transfer algorithm [17] to videos. Chen *et al*. [4] designed an RNN structure baseline and performed the warping operation in the feature domain. Gupta *et al*. [19] concatenated the former stylized frame with the current content frame before rendering and formed a flow loss as a constraint. Huang *et al*. [22] tried to integrate temporal coherence into the stylization network with a hybrid loss. Ruder *et al*. [40] extended their previous work [39] with new initializations and loss functions to improve robustness against large motions and strong occlusions. Temporal consistency can be improved with these optical flow constraints. However, optical flow estimation is not perfectly accurate, resulting in artifacts in the video results. Besides, it is computationally expensive, especially when the image size scales up. Considering these, another line of work tries to maintain the coherence of content inputs without using optical flow.

Li *et al*. [29] and Deng *et al*. [14] devised linear transformations for content features to preserve structure affinity. Liu *et al*. [33] used L1 normalization to replace the softmax operation of SANet [36] to get a more flat attention score

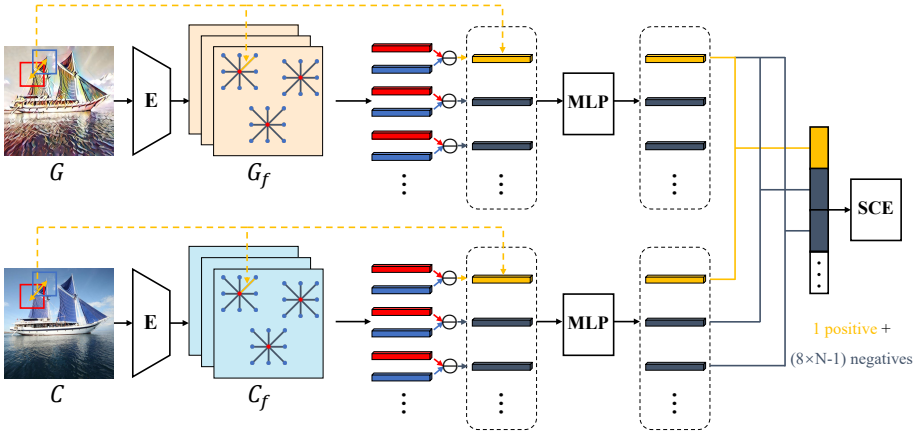


Fig. 3. Diagram of the proposed CCPL. C_f and G_f represent the encoded features from a specific layer of the encoder E. $\ominus$ denotes vector subtraction, and SCE means softmax cross-entropy. The yellow dashed lines illustrate how the positive pair is produced.

distribution. Wang *et al*. [46] proposed compound temporal regularization to enhance the robustness of the network to motions and illumination changes. Compared to these approaches, our proposed CCPL poses no requirements for the network architecture, making it exceptionally adaptive to other networks. With our SCTNet, the temporal consistency of video outputs surpasses SOTAs while the stylization remains satisfying. We also apply CCPL to other networks. The results show similar improvements in video stability (see Tab. 1).

Contrastive Learning. The original purpose of contrastive learning algorithms is to learn a good feature representation in a self-supervised scenario. A rich family of methods tried to achieve this by maximizing the mutual information of positive feature pairs while minimizing it in negative pairs [8,9,10,18,20,35]. Recent works extended contrastive learning to the field of image-to-image translation [37] and image style transfer [6]. Our work is most relevant to CUT [37] in using patch-based InfoNCE loss [35]. But CUT [37] utilized the correspondence of patches at the same locations for the image-to-image (Im2Im) translation task. However, our CCPL incorporates a neighbor-regulating scheme to preserve the correlations among neighboring patches, making it suitable for image and video generation. Besides, our experiment illustrates the effectiveness of CCPL on top of CUT [37] in the Im2Im translation task, as depicted in Sec. 4.4.

3 Methods

3.1 Contrastive Coherence Preserving Loss

Given two frames C_t and $C_{t+\Delta t}$ where Δt is the time interval in between, we assume the difference between the corresponding generated images G_t and $G_{t+\Delta t}$ is linearly dependent on the difference between C_t and $C_{t+\Delta t}$, when Δt is small:

$$\lim_{\Delta t \to 0} \mathcal{D}(C_{t+\Delta t}, C_t) \simeq \mathcal{D}(G_{t+\Delta t}, G_t), \tag{1}$$

where $\mathcal{D}(a, b)$ represents the difference between a and b. This constraint is probably too strict to hold for the whole image but technically sound for local patches where usually only simple image transformations, *e.g.*, translation or rotation, can occur. Under this assumption, we propose a generic Contrastive Coherence Preserving Loss (CCPL) applied to local patches to enforce this constraint. We show in Sec. 1 that our loss applied on neighboring patches is equivalent to that on corresponding patches of two frames, assuming Δt is small. Operating on a single frame frees us from processing multiple frames of a video source, saving computation budget.

To apply CCPL, first, we send the generated image G and its content input C to the fixed image encoder E to get feature maps of a specific layer, denoted as G_f and C_f (shown in Fig. 3). Second, we randomly sample N vectors[4] from G_f (red dots in Fig. 3), denoted as G_a^x where $x = 1, \cdots, N$. Third, we sample the *eight* nearest neighboring vectors of each G_a^x (blue dots in Fig. 3), denoted by $G_n^{x,y}$ where $y = 1, \cdots, 8$ is the neighbor index. Then, we accordingly sample from C_f at the same locations to get C_a^x and $C_n^{x,y}$, respectively. The differences between a vector and its neighboring vectors are measured by:

$$d_g^{x,y} = G_a^x \ominus G_n^{x,y}, \; d_c^{x,y} = C_a^x \ominus C_n^{x,y}, \tag{2}$$

where $\ominus$ represents vector subtraction. In order to realize Eq. 1, one simple thought is to enforce d_g equal to d_c. But in this case, an easy workaround of the network is to encourage G similar to C, meaning that this constraint would contradict the purpose of style transfer. Inspired by the recent progress in contrastive learning [8,20,35], we instead try to maximize the mutual information between "positive" difference vector pairs. A pair is only defined between a difference vector from C_f and G_f. Namely, the difference vectors of the same locations are defined as positive pairs between d_g and d_c, otherwise negative. The underlying intuition is also straightforward: the difference vectors of the same location should be most relevant in the latent space compared to other random pairs.

We follow the design of [8] to build a two-layer MLP (multi-layer perceptron) to map the difference vectors and normalize them onto a unit sphere before computing InfoNCE loss [35]. Mathematically:

$$L_{\text{ccp}} = \sum_{m=1}^{8\times N} -\log[\frac{\exp(d_g^m \cdot d_c^m/\tau)}{\exp(d_g^m \cdot d_c^m/\tau) + \sum_{n=1, n\neq m}^{8\times N} \exp(d_g^m \cdot d_c^n/\tau)}], \tag{3}$$

where τ stands for a temperature hyper-parameter set to 0.07 by default. With this setting, the temporal consistency of video outputs improves significantly (see Fig. 5 and Tab. 1) while the stylization remains satisfying or even gets better (see Fig. 6, Fig. 9, dirty texture disappears with our CCPL).

This loss avoids direct contradiction with style losses used to ensure style coherence between the generated and style image. Meanwhile, it can improve the temporal consistency of the generated video even without leveraging information from other frames of the input video. The complexity of CCPL is $\mathcal{O}\left(8 \times N\right)^2$,

[4] As encoded features are spatially decreased, each vector in the feature level corresponds to an image patch in the image level.

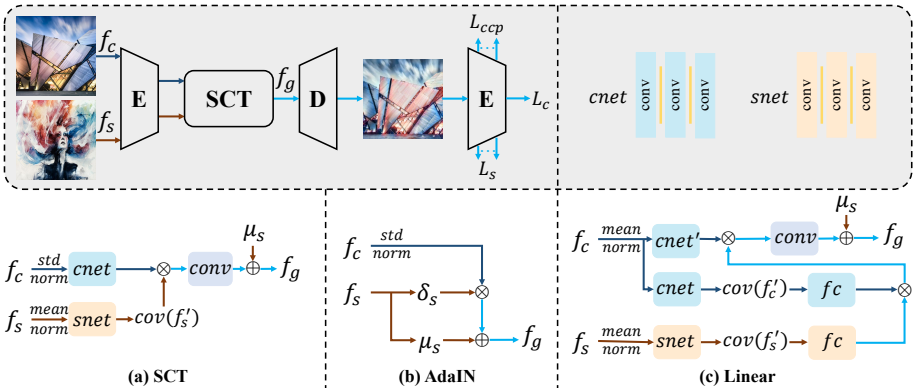


Fig. 4. Details of the proposed SCT module and its comparison with similar algorithms (AdaIN [23], Linear [29]). Here *conv* represents a convolutional layer, and the yellow lines in *cnet* and *snet* denote *relu* layers. Besides, *std norm* represents normalizing features by the means and standard deviations of channels, while *mean norm* normalizes features by the means of its channels.

where $8 \times N$ represents the number of sampled difference vectors. It is computationally affordable during training and has zero influence on inference speed (shown in Fig. 8*a*). CCPL can even work as a simple plugin to extend methods of other *image generation* tasks to produce videos with much better temporal consistency, as shown in Sec. 4.4.

3.2 Simple Covariance Transformation

With CCPL guaranteeing temporal consistency, our next goal is to design a simple and effective module for the fusion of content and style features for rich stylization. Huang *et al*. [23] proposed AdaIN to align channel-wise mean and variance of content and style features directly. Although simple enough, the inter-channel correlations are ignored, which are verified to be effective in the latter literature [14,29]. Li *et al*. [29] devised a channel-attention mechanism to transfer second-order statistics of style features onto corresponding content features. But we empirically find that the structure of Linear [29] can be simplified.

To combine the advantages of AdaIN [23] and Linear [29], we design a **Simple Covariance Transformation (SCT)** module to fuse style and content features. As shown in Fig. 4, first, we normalize the content feature f_c by the means and deviations of its channels [23] and the style feature f_s by the means of its channels [29] to get $\bar{f}_c$ and $\bar{f}_s$. To reduce computation costs, we send $\bar{f}_c$ and $\bar{f}_s$ to *cnet* and *snet* (*cnet* and *snet* both contain three convolutional layers, and two *relu* layers in between) to gradually reduce the dimension of channels ($512 \rightarrow 32$), and get $f_c^{'}$ and $f_s^{'}$. Then we flatten $f_s^{'}$ and calculate its covariance matrix $cov(f_s^{'})$ to find out the channel-wise correlations. After that, we simply fuse the features by performing a matrix multiplication between $cov(f_s^{'})$ and $f_c^{'}$ to get f_g. Finally, we use a single convolutional layer (denoted as *conv* in Fig. 4) to restore the channel dimension of f_g back to normal ($32 \rightarrow 512$) and add channel means of the original style feature before sending it to the decoder.

Combined with a symmetric encoder-decoder module, we name the whole network as **SCTNet**. The encoder is a VGG-19 network [43] pre-trained on ImageNet [13] to extract features from the content and style images, while the symmetric decoder needs to convert the fused feature back to images. Experiments suggest that our SCTNet is comparable to Linear [29] in stylization effects (see Fig. 6 and Tab. 1), while being lighter and faster (see Tab. 3).

3.3 Loss Function

Apart from the proposed CCPL, we adopt two commonly used losses [1,14,23,33] for style transfer. The overall training loss is a weighted sum of these three losses:

$$L_{\text{totoal}} = \lambda_c \cdot L_{\text{c}} + \lambda_s \cdot L_{\text{s}} + \lambda_{ccp} \cdot L_{\text{ccp}}. \tag{4}$$

The content loss L_c (the style loss L_s) is measured by the Frobenius norm of the differences between (means $\mu(\cdot)$ and standard deviations $\sigma(\cdot)$ of) the generated features and the content (style) features:

$$L_{\text{c}} = \left\| \phi_l(I_g) - \phi_l(I_c) \right\|_F, \tag{5}$$

$$L_{\text{s}} = \sum_l (\left\| \mu(\phi_l(I_g)) - \mu(\phi_l(I_s)) \right\|_F + \left\| \sigma(\phi_l(I_g)) - \sigma(\phi_l(I_s)) \right\|_F), \tag{6}$$

where $\phi_l(\cdot)$ denotes the feature map from the l-th layer of the encoder. For artistic style transfer, we use the features from $\{relu4\_1\}$, $\{relu1\_1,\ relu2\_1,\ relu3\_1,\ relu4\_1\}$, $\{relu2\_1,\ relu3\_1,\ relu4\_1\}$ to calculate the content loss, style loss, and CCPL, respectively. As for photo-realistic style transfer, we set the loss layers to $\{relu3\_1\}$, $\{relu1\_1,\ relu2\_1,\ relu3\_1\}$, $\{relu1\_1,\ relu2\_1,\ relu3\_1\}$ for the above losses. The loss weights are set to $\lambda_c = 1.0,\ \lambda_s = 10.0,\ \lambda_{ccp} = 5.0$ by default. Please check Sec. 4.3 for details about how we find these configurations.

4 Experiments

4.1 Experimental settings

Implementation details. We adopt content images from MS-COCO [32] data-set and style images from Wikiart [38] data-set to train our network. Both data-sets contain approximately 80,000 images. We use the Adam optimizer [26] with a learning rate of 1e-4 and the batch size of 8 to train the model for 160k iterations by default. During training, we first resize the smaller dimension of images to 512. Then we randomly crop 256×256 patches from images as the final input. For CCPL, we only treat difference vectors within the same content image as negative samples. More details are provided in the supplemental file.

Metrics. To comprehensively evaluate the performance of different algorithms and make the comparison fair, we adopt several metrics to assess the results' stylization effects and temporal consistency. To evaluate stylization effects, we compute *SIFID* [41] between the generated image and its style input to measure their style distribution distance. Lower SIFID represents closer style distributions

Table 1. Quantitative comparison of video and artistic style transfer. Here i stands for the interval of frames, and Pre. stands for *human preference score*. We show the *human preference score* of both artistic image style transfer (Art) and video style transfer (Vid) in the table. The results of *temporal loss* are magnified 100 times. We show the **first-place** score in bold and the second-place score with underlining.

Methods	SIFID (↓)	LPIPS(↓)		Temporal Loss (↓)		Pre. (↑)	
		i=1	i=10	i=1	i=10	Art	Vid
AdaIN [23]	2.44	0.184	0.444	5.16	7.92	0.028	0.028
AdaIN [23]+L_{ccp}	2.58	0.163	0.408	4.21	6.72	0.054	0.054
SANet [36]	2.40	0.227	0.478	6.31	13.72	0.062	0.046
SANet [36]+L_{ccp}	2.60	0.167	0.390	4.42	7.09	0.084	0.086
Linear [29]	2.38	0.160	0.417	4.25	7.61	0.076	0.080
Linear [29]+L_{ccp}	2.47	0.147	0.370	4.01	6.96	0.082	0.088
MCCNet [14]	2.34	0.162	0.424	4.21	7.64	0.088	<u>0.106</u>
AdaAttN [33]	2.48	0.207	0.419	4.87	6.49	<u>0.098</u>	0.094
DSTN [21]	2.83	0.234	0.450	5.72	10.76	0.070	0.038
IE [6]	2.99	0.182	0.379	4.35	6.76	0.054	0.058
ReReVST [46]	2.78	**0.137**	**0.359**	**2.97**	<u>5.19</u>	0.046	0.062
SCTNet	**2.29**	0.187	0.446	4.82	12.22	0.066	0.060
SCTNet+L_{nor}[11]	<u>2.31</u>	0.191	0.439	5.07	11.54	0.070	0.062
SCTNet+L_{ccp}	2.43	<u>0.144</u>	<u>0.367</u>	<u>3.45</u>	**5.08**	**0.122**	**0.138**

Table 2. Quantitative comparison of photo-realistic style transfer.

Metrics	Linear [29]	WCT^2 [47]	StyleNAS [2]	DSTN [21]	SCTNet	SCTNet+L_{ccp}
SIFID (↓)	1.82	1.86	2.37	3.35	**1.65**	2.14
LPIPS (↓)	0.395	0.419	0.379	0.464	0.427	**0.351**
Pre. (↑)	0.176	0.186	0.180	0.068	0.128	**0.262**

of a pair. To evaluate the visual quality and temporal consistency, we opt to *LPIPS* [48], which is originally used to measure the diversity of the generated images [12,24,27]. In our cases, small LPIPS represents few local distortions of the photo-realistic results or minor changes between two stylized video frames. Nonetheless, LPIPS only considers the correlations between stylized video frames while ignoring the changes between the original frames. As a supplement, we also adopt the *temporal loss* defined in [46] to measure temporal consistency. It is done by utilizing the optical flow between two frames to warp one stylized result and compute the Frobenius difference with another. We evaluate short-term (two adjacent frames) and long-term (9 frames in between) consistency for video style transfer. For short-term consistency, we directly use the ground-truth optical flow from the MPI Sintel data-set [3]. Otherwise, we use PWC-Net [44] to estimate the optical flow between two frames. The lower temporal loss represents better preservation of coherence between two frames.

For image style transfer comparison, we randomly choose 10 content images and 10 style images to synthesize 100 stylized images for each method and calculate their mean SIFID as the stylization metric. Besides, we compute the mean LPIPS to measure the visual quality of photo-realistic results. As for temporal

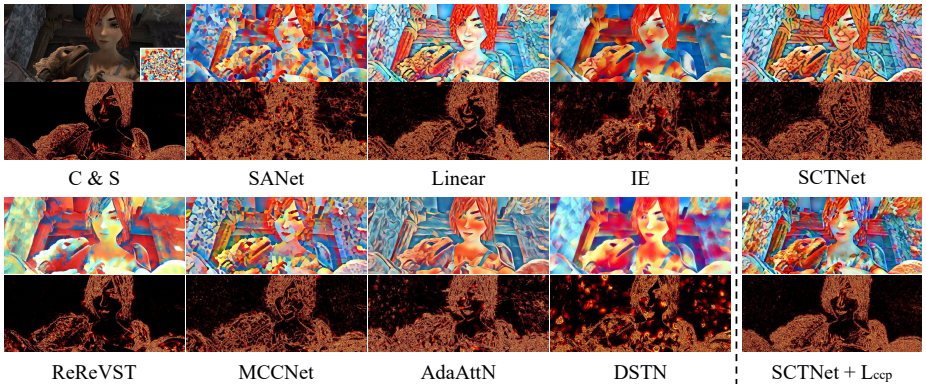


Fig. 5. Qualitative comparison of short-term temporal consistency. We compare our method with seven algorithms: SANet [36], Linear [29], IE [6], ReReVST [46], MCCNet [14], AdaAttN [33], DSTN [21]. The odd rows show the previous frames. The even rows show the heat-maps of differences between consecutive frames.

consistency, we randomly select 10 video clips (50 frames, 12 FPS each) from the MPI Sintel dataset [3] and use 10 style images to transfer these videos, respectively. Then we compute the mean LPIPS and temporal loss as the temporal consistency metrics. We also include human evaluation, which is more representative in image generation tasks. To do so, we invite 50 participants to choose their favorite stylized image/video from each image/video-style pair considering the visual quality, stylization effect, and temporal consistency. These participants come from different backgrounds, making the evaluation less biased towards a certain group of people. Overall, we get 500 votes for images and videos, respectively. Then we calculate the percentage of votes as the *human preference score*. All the evaluations are shown in Tab. 1 and Tab. 2.

4.2 Comparison with Former Methods

For video and artistic image style transfer, we compare our method with nine algorithms: AdaIN [23], SANet [36], DSTN [21], ReReVST [46], Linear [29], MCCNet [14], AdaAttN [33], IE [6], L_{nor} [11], which are the SOTAs of artistic image style transfer. Among these methods, [6,14,29,33] are also the most advanced single-frame-based video style transfer methods while ReReVST [46] is the SOTA multi-frames-based method. As for photo-realistic image style transfer, we compare our method with four SOTAs: Linear [29], WCT^2 [47], StyleNAS [2], DSTN [21]. Note that among all these mentioned algorithms, Linear [29] and DSTN [21] are most relevant to our method, since both of them are capable of transferring artistic and photo-realistic style onto images. We obtain all the test results from the official codes these methods provide.

Video style transfer. As shown in Tab. 1, our original SCTNet scores the best in SIFID, indicating its superiority in obtaining correct styles. Also, we can see the proposed CCPL improves the temporal consistency a lot with a minor decrease of the SIFID score, when the loss is applied to different methods. And our full model (with CCPL) exceeds all the single-frame methods [6,14,21,29,33,36]

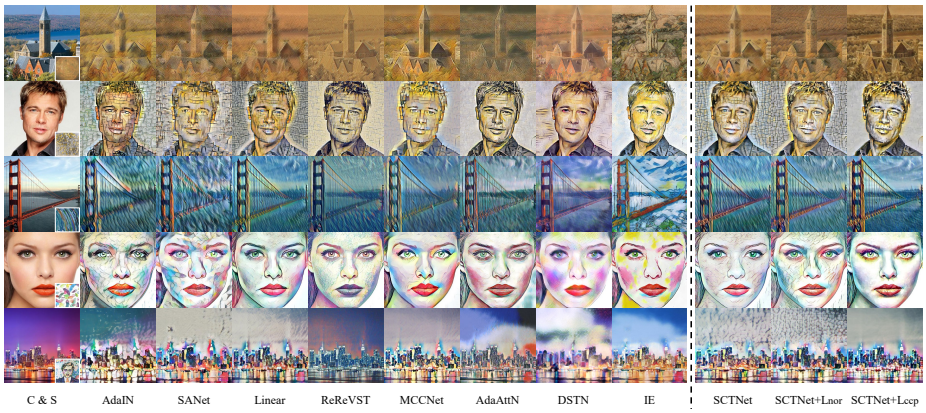


Fig. 6. Qualitative comparison of artistic style transfer. We compare our method with nine algorithms: AdaIN [23], SANet [36], Linear [29], ReReVST [46], MCCNet [14], AdaAttN [33], DSTN [21], IE [6], L_{nor} [11].

in both short-term and long-term temporal consistency, which are measured by LPIPS [48] and temporal loss, and performs on par with the SOTA multi-frame method: ReReVST [46]. However, our SIFID score exceeds ReReVST [46] significantly, which is consistent with the results shown in the qualitative comparison (See Fig. 6). The qualitative comparisons also show the advantage of our CCPL in maintaining short-term (Fig. 5) temporal consistency of the original video as our heat-map difference is mostly similar to ground-truth. We have another figure in the supplemental file to show the comparison of long-term temporal consistency. In terms of human preference score, our full model also ranks the best, further validating the effectiveness of our CCPL.

Artistic style transfer. As shown in Fig. 6, AdaIN [23] generates results with severe shape distortion (*e.g.*, house in the 1^{st} and bridge in the 3^{rd} row) and disarranged texture patterns (4^{th}, 5^{th} rows). SANet [36] also has shape distortion and misses some structural details in its results ($1^{st} \rightarrow 3^{rd}$ rows). Linear [29] and MCCNet [14] have relatively quite clean outputs. However, Linear [29] loses some content details ($1^{st}, 3^{rd}$ rows), and some results of MCCNet [14] have checkerboard artifacts in local regions (around collar in the 2^{nd} row and corner of mouth in the 4^{th} row). ReReVST [46] shows obvious color distortion ($2^{nd} \rightarrow 5^{th}$ rows). AdaAttN [33] is effective in reducing messy textures but the stylization effect seems to degenerate in some cases (1^{st} row). The results of DSTN [21] have severe obvious distortion ($3^{rd}, 4^{th}$ rows). And the results of IE [6] are less similar to the original style ($1^{st}, 3^{rd}, 5^{th}$ rows). Our original SCTNet captures accurate style ($2^{nd}, 3^{rd}$ rows), but there are some messy regions in the generated images as well ($4^{th}, 5^{th}$ rows). When adding L_{nor} [11], some results are even messier ($4^{th}, 5^{th}$ rows). However, with CCPL, the generated results of our full model maintain well the structures of their content sources with vivid and appealing colorization. Besides, this effect is reinforced by its multi-level scheme. Therefore, irregular textures and local color distortions are decreased significantly. It even helps to improve stylization with better preservation of the semantic information of the content sources (as shown in Fig. 9).

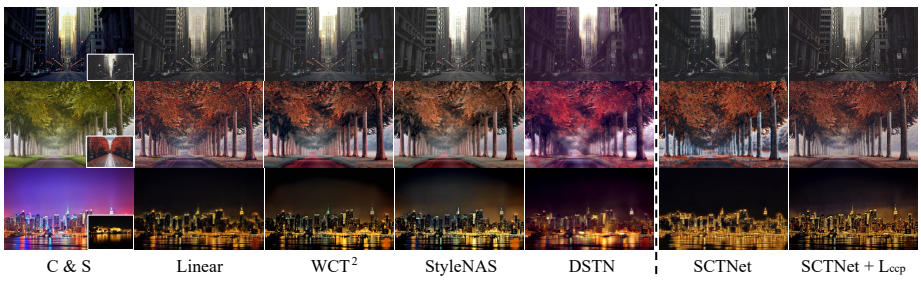


Fig. 7. Qualitative comparison of photo-realistic style transfer. We compare our method with four algorithms: Linear [29], WCT^2 [47], StyleNAS [2] and DSTN [21].

Photo-realistic style transfer. Since CCPL can preserve the semantic information of the content source and significantly reduce local distortions, it is well-suited for the task of photo-realistic style transfer. We make slight changes to SCTNet to enable it for this task: build a shallower encoder by throwing off layers beyond $relu3\_1$, then use feature maps from all three layers to calculate CCPL. As shown in Fig. 7, Linear [29] and DSTN [21] generates results with detail losses (vanished windows in the 3^{rd} row). As for WCT^2 [47] and StyleNAS [2], some results of them show unreasonable color distribution (red road in the 2^{nd} row). In comparison, our full model generates results comparable or even better than those SOTAs, with high visual quality and appropriate stylization, which is consistent with the quantitative comparison shown in Tab. 2.

Efficiency analysis. Our model is quite efficient due to the simple feed-forward architecture of the network and the efficient feature fusion module SCT. We use a single 12GB Titan XP GPU with no other ongoing programs to compare its running speed with other algorithms. Tab. 3 shows the average running speed (over 100 independent runs) of different methods on three input image scales. The result suggests that SCTNet surpasses the SOTAs in efficiency at different scales (comparisons for photo-realistic style transfer methods are provided in the supplemental file), indicating the feasibility of our algorithm for real-time use.

4.3 Ablation Studies

There are several factors relevant to the performance induced by the CCPL: 1) layers to apply the loss; 2) the number of difference vectors sampled each layer; 3) the loss weight ratio with the style loss. Therefore, we conduct several experiments by enumerating the number of CCPL layers from 0 to 4 (start from the deepest layer) and choosing from [16, 32, 64, 128] as the number of sampled combinations to show the impacts of the first two factors. Then we adjust the loss weight ratio between the CCPL and the style loss to manifest which ratio gives the best trade-off between style effects and temporal coherence. To be noted, the stylization score here represents the SIFID score, and the temporal consistency is measured by: $(20 - 10 \times \text{LPIPS} - \text{temporal loss})$ to show the escalating trend.

From the sub-figures, we can see that, as the number of CCPL layers increases, the short-term (Fig. 8*d*) and long-term (Fig. 8*e*) temporal consistency increases with the reduction of stylization score (Fig. 8*b*) and greater computation (Fig. 8*a*). And when the number of CCPL layers increases from 3 to 4, the

Table 3. Execution speed comparison (unit: FPS). We use a single 12GB Titan XP GPU for all the execution time testing. OOM denotes the Out-Of-Memory error.

Artistic	Ad [23]	SA [36]	LT [29]	Re [46]	MC [14]	AN [33]	DN [21]	IE [6]	Ours
256 × 256	40.0	34.5	66.7	37.0	22.2	15.6	15.9	31.3	**77.0**
512 × 512	12.5	14.3	18.9	13.7	8.1	12.5	4.2	13.0	**21.7**
1024 × 1024	2.7	2.7	4.6	2.8	1.9	2.1	1.2	2.6	**5.0**

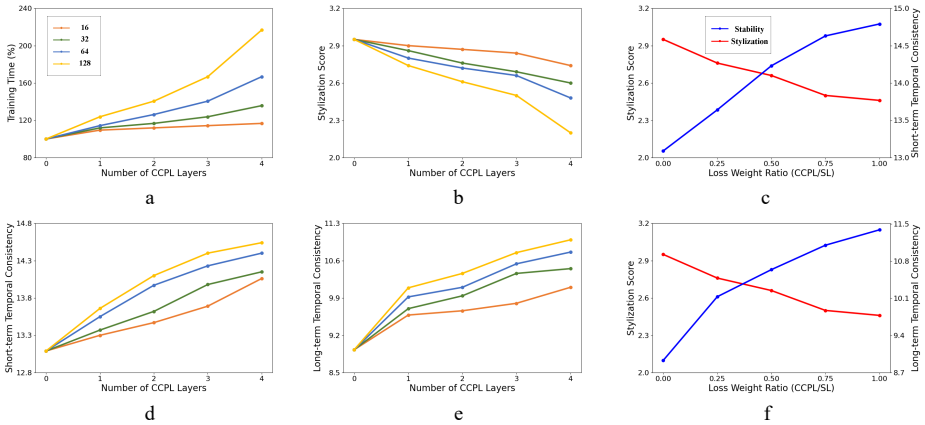


Fig. 8. Ablation studies on three factors of the CCPL: 1) layers to apply the loss; 2) the number of vectors sampled each layer; 3) the loss weight ratio with style loss.

changes of temporal consistency are minor. In contrast, the computation costs increase significantly, and the stylization effects are much weaker. Therefore, we choose 3 as the default setting for the number of CCPL layers.

As for the number of sampled difference vectors (per layer), the blue lines (64 sampled vectors) in Fig. 8*d* & *e* are near the yellow lines (128 sampled vectors), which means the performance of these two settings are close on improving temporal consistency. However, sampling 128 difference vectors per layer brings a significantly heavier computation burden and style degeneration. So we sample 64 difference vectors per layer by default.

The loss weight ratio can also be regarded as a handle to adjust temporal consistency and stylization. Fig. 8*c* & *f* show the trade-off between temporal consistency and stylization when the loss weight ratio changes. We find 0.5 a good choice for the weight ratio because it gives a good trade-off between temporal consistency improvement and stylization score reduction. We show the qualitative results of ablation studies on CCPL in the supplemental file and more analysis, such as different sampling strategies in CCPL.

4.4 Applications

CCPL on existing methods. CCPL is highly flexible and can be plugged into other methods with minor modifications. We apply the proposed CCPL on three typical former methods: AdaIN [23], SANet [36], Linear [29]. All these methods achieve consistent improvements in temporal consistency with only a

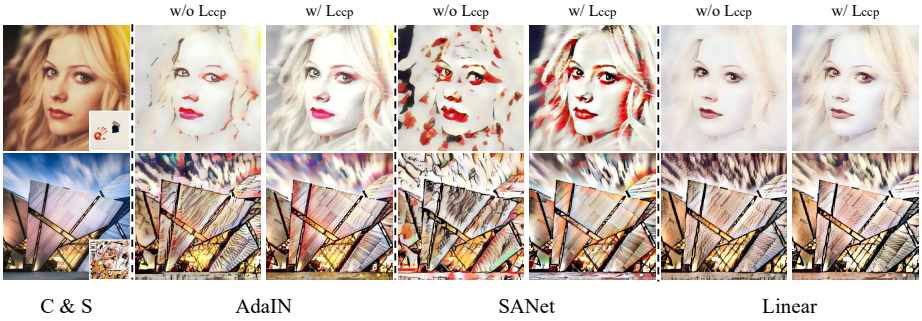
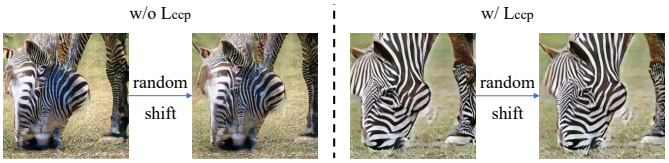


Fig. 9. CCPL can be easily applied to other methods, such as AdaIN [23], SANet [36] and Linear [29], to improve visual quality.



	w/o Lccp	w/ Lccp
LPIPS	0.470	0.442
Temp loss	0.105	0.075

Fig. 10. Comparison of applying the CCPL on CUT [37] with its original model.

slight decrease on the SIFID score (see Tab. 1 and Fig. 9). The result reveals the effectiveness and flexibility of the CCPL.

Image-to-image translation. CCPL can be easily added to other generation tasks like image-to-image translation. We apply our CCPL on a recent image-to-image translation method CUT [37] and then train the model with the same horse2zebra dataset. The results in Fig. 10 demonstrate that our CCPL improves both the visual quality and temporal consistency. Please refer to the supplemental file for more applications.

5 Conclusions

In this work, we propose CCPL to preserve content coherence during style transfer. By contrasting the feature differences of image patches, the loss encourages the difference of patches of the same location in the content and generated images to be similar. Models trained with CCPL achieve a good trade-off between temporal consistency and style effects. We also propose a simple and effective module for aligning second-order statistics of the content feature with style feature. Combining these two techniques, our full model is light and fast while generating satisfying image and video results. Besides, we demonstrate the effectiveness of the proposed loss on other models and tasks, such as image-to-image style transfer, which shows the vast potential of our loss for broader applications.

Acknowledgements This work was supported by the National Natural Science Foundation of China 62192784.

References

1. An, J., Huang, S., Song, Y., Dou, D., Liu, W., Luo, J.: Artflow: Unbiased image style transfer via reversible neural flows. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 862–871 (2021)
2. An, J., Xiong, H., Ma, J., Luo, J., Huan, J.: Stylenas: An empirical study of neural architecture search to uncover surprisingly fast end-to-end universal style transfer networks. arXiv preprint arXiv:1906.02470 (2019)
3. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European conference on computer vision. pp. 611–625. Springer (2012)
4. Chen, D., Liao, J., Yuan, L., Yu, N., Hua, G.: Coherent online video style transfer. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1105–1114 (2017)
5. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: Stylebank: An explicit representation for neural image style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1897–1906 (2017)
6. Chen, H., Wang, Z., Zhang, H., Zuo, Z., Li, A., Xing, W., Lu, D., et al.: Artistic style transfer with internal-external learning and contrastive learning. Advances in Neural Information Processing Systems **34** (2021)
7. Chen, T.Q., Schmidt, M.: Fast patch-based style transfer of arbitrary style. arXiv preprint arXiv:1612.04337 (2016)
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
9. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
10. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021)
11. Cheng, J., Jaiswal, A., Wu, Y., Natarajan, P., Natarajan, P.: Style-aware normalized loss for improving arbitrary style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 134–143 (2021)
12. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8188–8197 (2020)
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
14. Deng, Y., Tang, F., Dong, W., Huang, H., Ma, C., Xu, C.: Arbitrary video style transfer via multi-channel correlation. arXiv preprint arXiv:2009.08003 (2020)
15. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. arXiv preprint arXiv:1610.07629 (2016)
16. Gao, C., Gu, D., Zhang, F., Yu, Y.: Reconet: Real-time coherent video style transfer network. In: Asian Conference on Computer Vision. pp. 637–653. Springer (2018)
17. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
18. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)
19. Gupta, A., Johnson, J., Alahi, A., Fei-Fei, L.: Characterizing and improving stability in neural style transfer. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4067–4076 (2017)

20. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
21. Hong, K., Jeon, S., Yang, H., Fu, J., Byun, H.: Domain-aware universal style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14609–14617 (2021)
22. Huang, H., Wang, H., Luo, W., Ma, L., Jiang, W., Zhu, X., Li, Z., Liu, W.: Real-time neural style transfer for videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 783–791 (2017)
23. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
24. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European conference on computer vision (ECCV). pp. 172–189 (2018)
25. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
27. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-image translation via disentangled representations. In: Proceedings of the European conference on computer vision (ECCV). pp. 35–51 (2018)
28. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE transactions on pattern analysis and machine intelligence **30**(2), 228–242 (2007)
29. Li, X., Liu, S., Kautz, J., Yang, M.H.: Learning linear transformations for fast image and video style transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3809–3817 (2019)
30. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. arXiv preprint arXiv:1705.08086 (2017)
31. Li, Y., Liu, M.Y., Li, X., Yang, M.H., Kautz, J.: A closed-form solution to photorealistic image stylization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 453–468 (2018)
32. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
33. Liu, S., Lin, T., He, D., Li, F., Wang, M., Li, X., Sun, Z., Li, Q., Ding, E.: Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6649–6658 (2021)
34. Luan, F., Paris, S., Shechtman, E., Bala, K.: Deep photo style transfer. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4990–4998 (2017)
35. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
36. Park, D.Y., Lee, K.H.: Arbitrary style transfer with style-attentional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5880–5888 (2019)
37. Park, T., Efros, A.A., Zhang, R., Zhu, J.Y.: Contrastive learning for unpaired image-to-image translation. In: European Conference on Computer Vision. pp. 319–345. Springer (2020)
38. Phillips, F., Mackintosh, B.: Wiki art gallery, inc.: A case for critical thinking. Issues in Accounting Education **26**(3), 593–608 (2011)

39. Ruder, M., Dosovitskiy, A., Brox, T.: Artistic style transfer for videos. In: German conference on pattern recognition. pp. 26–36. Springer (2016)
40. Ruder, M., Dosovitskiy, A., Brox, T.: Artistic style transfer for videos and spherical images. International Journal of Computer Vision **126**(11), 1199–1219 (2018)
41. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4570–4580 (2019)
42. Sheng, L., Lin, Z., Shao, J., Wang, X.: Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8242–8250 (2018)
43. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
44. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8934–8943 (2018)
45. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: Feed-forward synthesis of textures and stylized images. In: ICML. vol. 1, p. 4 (2016)
46. Wang, W., Yang, S., Xu, J., Liu, J.: Consistent video style transfer via relaxation and regularization. IEEE Transactions on Image Processing **29**, 9125–9139 (2020)
47. Yoo, J., Uh, Y., Chun, S., Kang, B., Ha, J.W.: Photorealistic style transfer via wavelet transforms. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9036–9045 (2019)
48. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)