Pattern Analysis and Applications (2023) 26:343–355
https://doi.org/10.1007/s10044-022-01106-y

THEORETICAL ADVANCES

Correlation-based and content-enhanced network for video style transfer

Honglin Lin[1] · Mengmeng Wang[1] · Yong Liu[1] · Jiaxin Kou[2]

Received: 9 February 2022 / Accepted: 27 August 2022 / Published online: 18 September 2022

Abstract
Artistic style transfer aims to migrate the style pattern from a referenced style image to a given content image, which has achieved significant advances in recent years. However, producing temporally coherent and visually pleasing stylized frames is still challenging. Although existing works have made some effort, they rely on the inefficient optical flow or other cumbersome operations to model spatiotemporal information. In this paper, we propose an arbitrary video style transfer network that can generate consistent results with reasonable style patterns and clear content structure. We adopt multi-channel correlation module to render the input images stably according to cross-domain feature correlation. Meanwhile, Earth Movers' Distance is used to capture the main characteristics of style images. To maintain the semantic structure during the stylization, we also employ the AdaIN-based skip connections and self-similarity loss, which can further improve the temporal consistency. Qualitative and quantitative experiments have demonstrated the effectiveness of our framework.

Keywords Video style transfer · Multi-channel correlation · Earth movers' distance · content-enhanced

1 Introduction

Artistic style transfer is an attractive technique that aims to migrate the desired style pattern from an exemplar style image to an input content image, and has gained growing interest in the computer vision community. The seminal work by Gatys et al. [1] first showed that the Gram matrix of features extracted from a pre-trained image classification network [2] can represent the visual style of an image. Since then, numerous methods have been developed to address this interesting problem, from optimization-based single style models to feed-forward arbitrary style models which have greatly improve the efficiency, diversity, robustness and stylization quality.

However, the naive extension from image to video may produce severe flickering effects between adjacent frames. Such artifacts mainly come from the lack of spatiotemporal information during the training process, so the same semantic area would be rendered into different appearances. Ruder et al. [3] added temporal consistency loss based on Gatys' method by tracking pixels with pre-computed optical flow. Feed-forward models were later proposed to speed up the optimization process [4–6]. Chen et al. [7] designed two sub-networks to do the warping in the feature space dynamically. Although these methods can generate smoothed results, they depend highly on the optical flow. If the estimated optical flow is not accurate enough, ghosting artifacts would appear at the motion boundaries of objects, and the heavy computation costs largely limit the model's practical applications. Besides, there is always a trade-off between stylization strength and temporal consistency [8], and sometimes the output image loses the richness of stylized details.

Therefore, the key to addressing the above problems is to design a more stable model instead of adding extra explicit or implicit constraints. Recently, Deng et al. [9] proposed the Multi-Channel Correlation (MCC) module,

✉ Yong Liu
yongliu@iipc.zju.edu.cn

Honglin Lin
linhl97@zju.edu.cn

Mengmeng Wang
mengmengwang@zju.edu.cn

Jiaxin Kou
jiaxin.kjx@alibaba-inc.com

[1] Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Engineering, Zhejiang University, Hangzhou, China

[2] Data Technology and Product Department-Platform Technology, Alibaba Group, Hangzhou, China

Fig. 1 Our network can generate consistent stylized frames with vivid style patterns and clear content structure

which can stylize the input image with reasonable style patterns while keeping consistency among frames. Employing it as the basic transfer module, we propose a new video style transfer framework with better performance in this paper. First, we adopt the Earth Movers' Distance loss to measure distribution differences between features. Thus the model can migrate the main characteristics of the style image and prevent the degradation effect in the trade-off mentioned above. Then, we add the AdaIN-based skip connections and self-similarity loss to ensure well-preserved content structure and clear layout after stylization. Besides, they can further improve the temporal consistency by removing the unstable texture. Generally speaking, our main contributions can be summarized as follows:

1. We combine the multi-channel correlation module with Earth Movers' Distance loss to generate coherent stylized results with reasonable distribution of style patterns.
2. We combine the AdaIN-based skip connections and self-similarity loss to enhance the content structure and capture more stylized details, further improving the consistency.
3. Qualitative and quantitative experiments demonstrate the effectiveness of our model, which has state-of-the-arts temporal stability and excellent stylization quality (shown in Fig. 1).

The rest of the paper is organized as follows: Sect.2 reviews related work in image and video style transfer. Section 3 introduces the proposed framework in detail. Experimental results and analysis are presented in Sect. 4. Finally, we conclude the paper in Sect. 5.

2 Related work

2.1 Image style transfer

With the rapid development of deep learning, significant advances have been made in artistic style transfer in the past few years. Neural Style Transfer (NST) [1] first formulated style as the Gram matrix of the feature maps extracted from pre-trained image classification networks [2]. However, it takes several minutes to stylize an image due to the optimization process. Johnson et al. [10] achieved real-time style transfer by training feed-forward network with style loss and perceptual loss proposed by NST. But their model can only represent one style at a time. Chen et al. [11] shared encoder and decoder across multiple styles and stored the filters corresponding to each style in a bank layer. Dumoulin et al. [12] proposed conditional instance normalization (CIN) layer, embedding styles into affine parameters. Inspired by CIN, Huang et al. [13] proposed the milestone arbitrary style transfer algorithm AdaIN. AdaIN normalized the content feature and then aligns its mean and variance with the style feature. Li et al. [14] proposed multi-scale whitening and coloring transformation (WCT) to match the second-order statistics directly. LST [15] learned a linear transformation matrix by two light-weighted convolutional neural networks (CNNs) to replace WCT operation, which is much more efficient and flexible. However, these holistic transformations methods still lead to unsatisfactory local style patterns. Therefore, SANet [16] and AAMS [17] integrated the local style patterns according to the semantic spatial distribution of the content image via self-attention mechanism. Recently, Liu et al. [18] designed a novel module to

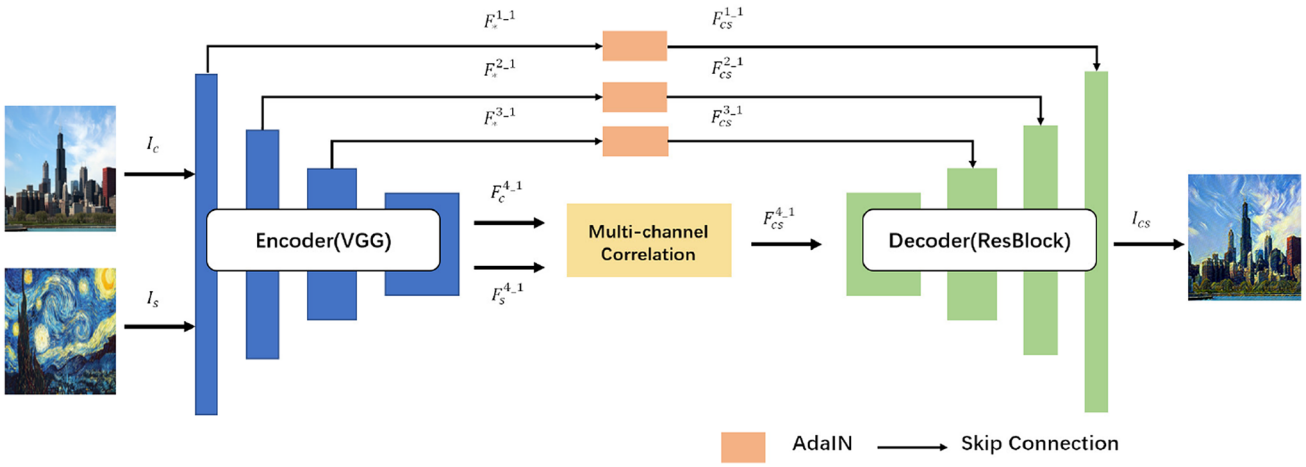

Fig. 2 Overall structure of our model. It is consists of the VGG-19 encoder and the ResBlock-styled decoder, with the AdaIN-based skip connections at multiple levels and the multi-channel correlation module at the bottleneck

adaptively perform attentive normalization on per-point basis, and extend their model with slight modifications to achieve state-of-the-art video style transfer.

Although producing impressive results, most methods are still unsuitable for video sequence because they generate severe flickering artifacts. As stated in [4], since the image model processes video frame by frame, the slight variations between adjacent frames would be amplified into different stylized appearances, which inevitably created such artifacts.

2.2 Video style transfer

Existing video style transfer methods can be roughly divided into two categories.

The first category is warping previous frame to the current through optical flow to form temporal consistency loss [3]. Huang et al. [4] integrated the constraint into feed-forward network to accelerate the inference. Gupta et al. [5] revealed that the trace of the Gram matrix is inversely related to the stability of the model and adopted the recurrent neural network (RNN). Lai et al. [19] and Gao et al. [20] both combined CNN with long short-term memory (LSTM) that is more expressive than RNN. Wang et al. [8] generated random optical flow through Gaussian sampling and warped the single image to simulate its adjacent frames, thereby ingeniously obtain a "video dataset" for training.

The second category is adding extra sub-networks to estimate the optical flow and motion mask dynamically and then warping sequentially in the feature (or image) space [7]. Vid2vid [21] further used discriminator to improve the accuracy of the sub-networks. Zhou et al. [22] regarded the flickering artifacts as high-frequency noise, and designed a parallel branch to generate the temporal denoising mask. These methods explicitly perform alignment, producing much more consistent result than the first type methods. However, the optical flow is also needed during inference, resulting in poor efficiency.

Recently, Deng et al. [9] revisited the self-attention mechanism, and proposed the MCCNet for temporal coherent video style transfer that does not involve the calculation of optical flow. In this work, we adopt it as basic transfer module.

3 Method

As shown in Fig. 2, our model takes a content image I_c and a style image I_s as input and synthesize a stylized image I_{cs}. It is based on the widely-used encoder-decoder paradigm, with the AdaIN-based skip connections between them and the multi-channel correlation module at the bottleneck. In this section, we will introduce all components of our model and loss functions used in training.

3.1 Encoder and decoder

Following previous works, we employ the pre-trained VGG-19 [2] network as the encoder.[1] VGG-19 is a simple and intuitive architecture stacked by multiple small convolutional blocks. Given the input RGB image pair I_c and I_s, we first scale them into the same size, and use the encoder to extract their multi-resolutions feature maps, respectively. We denote the extracted feature of layer *ReLUx_1* in VGG-19

[1] The pre-trained VGG-19 weights can be downloaded at https://drive.google.com/file/d/1EpkBA2K2eYILDSyPTt0fztz59UjAIpZU/view.

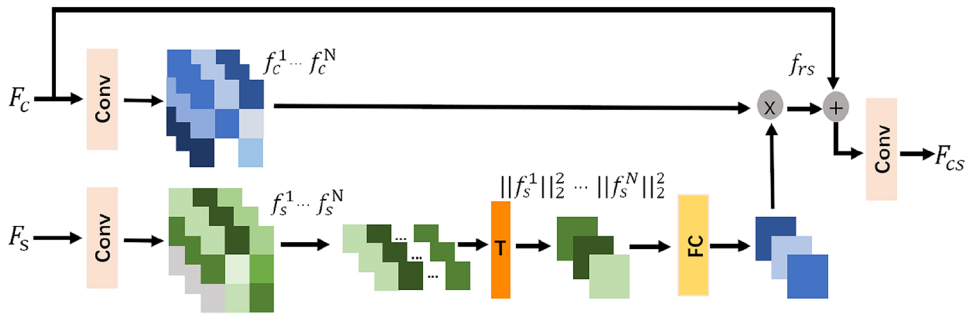

Fig. 3 Multi-channel correlation module [9]

as $F_*^{x\_1}$, where $*$ can be c or s here representing content and style features, respectively. Then we fed $F_*^{x\_1}(x = 1 \cdots 3)$ into AdaIN module and $F_*^{4\_1}$ into multi-channel correlation module, producing stylized features $F_{cs}^{x\_1}(x = 1 \cdots 4)$. Finally, the decoder converts the stylized features into the output image I_{cs}. A common choice for decoder is the symmetric structure of VGG-19 in style transfer. However, we employ residual blocks [23] to constitute the decoder to make it more expressive and integrate the stylized features from skip connections. More implementation details of the network are presented in Appendix A.

3.2 Style transfer component

3.2.1 Multi-channel correlation

Since its great success in natural language processing [24], self-attention mechanism has been introduced into artistic style transfer by Park et al. [16] and Yao et al. [17]. However, they ignored the inter-channel relationship of feature maps, attentively fused features via only one spatial mask. Recently, Deng et al. [9] revisited the self-attention mechanism and proposed the Multi-Channel Correlation (MCC) module, as shown in Fig. 3. Since each channel of feature maps usually represents different semantics like colors, textures, shapes, and other abstract patterns, the channel-wise operation is more reasonable. We employ it as the basic transfer module, which is explained in the following.

MCC module takes extracted features $F_*^{4\_1} \in \mathbb{R}^{C\times H\times W}$ as input, where C, H, W are number of channels (namely dimensions), height and width of feature maps. For channel i, the content feature and style feature can be reshaped into row vector $f_c^i \in \mathbb{R}^{1\times N}$, $f_c^i = [c_1, c_2, \cdots, c_N]$ and $f_s^i \in \mathbb{R}^{1\times N}$, $f_s^i = [s_1, s_2, \cdots, s_N]$, where $N = H \times W$. Then the correlation matrix is calculated by their dot product:

$$CO^i = f_c^{iT} \otimes f_s^i \tag{1}$$

$CO^i \in \mathbb{R}^{N\times N}$ measures the semantic similarity spatially. So we can rearrange the style feature, namely assigning different weights to each element in f_s^i according to CO^i. Then the style feature f_s^i can be integrated into the content feature f_c^i properly, which is formulated as:

$$f_{cs}^i = f_c^i + f_s^i \otimes CO^{iT} = \left(1 + \left\|f_s^i\right\|_2^2\right) f_c^i \tag{2}$$

where $\left\|f_s^i\right\|_2^2 = \sum_{j=1}^N s_j^2$, $f_{cs}^i \in \mathbb{R}^{1\times N}$ is the i-th channel stylized feature.

For better stylization quality, MCC module further calculates the correlation between each content channel and every style channel and then weighted them together as:

$$f_{cs}^i = \left(1 + \sum_{k=1}^{C} w_k \left\|f_s^k\right\|_2^2\right) f_c^i \tag{3}$$

where w_k is a real number representing the weights of the k-th channel, which is learned by fully connected layers during training. Compared with most attention mechanisms, MCC module does not use *Softmax*, *Sigmoid* or other nonlinear functions to normalize the attention weights. The whole pipeline can be approximately regarded as performing a linear transformation on the input content feature $F_c^{4\_1}$ to produce the stylized feature $F_{cs}^{4\_1}$. As stated in [15], linear transformation is capable of preserving the feature affinity, which means the dense pair-wise relations of pixels are preserved well after stylization. We believe this enables the MCC module to be robust to tiny variations and avoid violent changes among frames. Thus, the coherence of input frames can be naturally migrated to output frames.

3.2.2 Earth movers' distance loss

Earth Movers' Distance (EMD) measures the similarity between two probability distributions, also referred to as Wasserstein Distance. In 2017, Arjovsky et al. [25] proposed the famous Wasserstein generative adversarial networks (WGAN), significantly improving GAN's training stability and avoiding mode collapse. Compared to JS divergence, EMD is much more continuous, and can still represent distances when the probability distributions do not overlap, thus providing meaningful gradients to the generator even in this case. Following the idea that the

essence of style transfer is to align feature distributions [26], we add EMD loss to improve the stylization quality further. However, the original EMD has a time complexity in the order $O(n^3)$ to find the optimal transport matrix. To speed up the optimization, we use relaxed Earth Mover Distance (rEMD) as derived in [27], and the rEMD loss is formulated as:

$$\mathcal{L}_{r\text{EMD}} = \max\left(\frac{1}{HW}\sum_{i=1}^{HW}\min_j C_{ij}, \frac{1}{HW}\sum_{j=1}^{HW}\min_i C_{ij}\right) \tag{4}$$

where the cost of transport C_{ij} is calculated by cosine distance:

$$C_{ij} = 1 - \frac{F_{s,i}^{x\_1} \cdot F_{cs,j}^{x\_1}}{\left\|F_{s,i}^{x\_1}\right\|\left\|F_{cs,j}^{x\_1}\right\|} \tag{5}$$

3.3 Content-enhanced component

3.3.1 AdaIN-based skip connections

Skip connection is a simple but effective method widely used in image segmentation and image generation tasks. It introduces the shallow feature on the corresponding scale into the upsample process while skipping the intermediate module. Therefore, the decoder can acquire more low-level information lost in downsample to generate refined results. Besides, it can make the structure of the output image much clearer. We believe that this will reduce the uncertainty during stylization, especially in areas near the semantic boundary. But in style transfer, the features in the decoder are already aligned with the style domain. If we directly merge them with the content domain features from the encoder, it will increase the difficulty of network training and produce unnatural artifacts. To address this problem, we add AdaIN [13] module on the skip connections, which aligns the mean and variance of the content feature with those of the style feature efficiently. Taking the content feature $F_c^{x\_1}$ and the style feature $F_s^{x\_1}$ as input, AdaIN can be formulated as:

$$\text{AdaIN}(F_c^{x\_1}, F_s^{x\_1}) = \sigma(F_s^{x\_1})\left(\frac{F_c^{x\_1} - \mu(F_c^{x\_1})}{\sigma(F_c^{x\_1})}\right) + \mu(F_s^{x\_1}) \tag{6}$$

where $\mu(\cdot) \in R^C$, $\sigma(\cdot) \in R^C$ are vectors representing the mean and standard variance of each channel, respectively. Through the multi-level AdaIN-based skip connections, the output image would contain rich stylized details, such as color, texture, brushstroke, etc.

3.3.2 Self similarity loss

Correlation matrix describes the statistical relationship between random variables, a primary metric in data analysis. For feature maps, it reflects the relative relationship of spatial elements, that is, how they are combined, representing the semantic structure of the image to some extent. Liu et al. [18] and Xu et al. [28] both designed the coherence loss by calculating the cross-frame similarity, but their methods need to finetune on the video dataset. We use self similarity loss [27] instead of cross-frame similarity loss, for it can prevent excessive stylization from damaging the semantic structure of the image and is easy to implement. In addition, it can remove the dirty and unstable texture that existed in the original smooth area after stylization, further improving the temporal coherence. The self similarity loss is formulated as:

$$\mathcal{L}_{\text{self}-\text{sim}} = \frac{1}{(HW)^2}\sum_{i,j}\left|\frac{D_{ij}^{c}}{\sum_i D_{ij}^{c}} - \frac{D_{ij}^{cs}}{\sum_i D_{ij}^{cs}}\right| \tag{7}$$

where D^c and D^{cs} represent the correlation matrix of the content feature $F_c^{x\_1}$ and the stylized feature $F_{cs}^{x\_1}$, respectively, measuring the similarity between the feature vector of each spatial position. To compute D^*, we first reshape $F_* \in \mathbb{R}^{C\times H\times W}$ into $F_* \in \mathbb{R}^{C\times HW}$, and perform the following matrix multiplication (we omit superscript for convenience):

$$D^* = F_*^T \otimes F_* \tag{8}$$

3.4 Training loss

3.4.1 Style loss

In addition to rEMD loss, we employ the commonly used mean-variance matching loss because cosine distance in Eq. 5 ignores features' magnitude, leading to unpleasant artifacts.

$$\begin{aligned}\mathcal{L}_{\text{sty}} = \sum_{x=1}^{4} &\left\|\mu\left(F_{cs}^{x\_1}\right) - \mu\left(F_s^{x\_1}\right)\right\|_2 \\ &+ \left\|\sigma\left(F_{cs}^{x\_1}\right) - \sigma\left(F_s^{x\_1}\right)\right\|_2\end{aligned} \tag{9}$$

3.4.2 Content loss

Like previous work, perceptual loss is necessary for plausible results:

$$\mathcal{L}_{\text{cont}} = \sum_{x=1}^{4}\left\|F_c^{x\_1} - F_{cs}^{x\_1}\right\|_2 \tag{10}$$

3.4.3 Illumination loss

The illumination may change slightly in adjacent frames because of camera motion, occlusion or other factors. These subtle variations in video frames could result in severe flicking artifacts. Following [9], we add random Gaussian noise to simulate these variations. The illumination loss is formulated as:

$$\mathcal{L}_{\text{illum}} = \left\| F(I_c, I_s) - F(I_c + \Delta, I_s) \right\|_2 \tag{11}$$

where $F(\cdot)$ represents the entire stylization process of our model, $\Delta \sim \mathcal{N}(0, \sigma^2 I)$. With illumination loss, the network can be more robust to complex light conditions in input videos.

3.4.4 Identity loss

We also employ the identity loss in [16] to maintain the content structure without losing the richness of the style patterns. The identity loss is formulated as:

$$\mathcal{L}_{\text{ide1}} = \|I_{cc} - I_c\|_2 + \|I_{ss} - I_s\|_2 \tag{12}$$

$$\mathcal{L}_{\text{ide2}} = \|F_{cc} - F_c\|_2 + \|F_{ss} - F_s\|_2 \tag{13}$$

where I_{cc} (I_{ss}) denotes the stylized image using a natural image(an artistic painting) as content image and style image. F_{cc} (F_{ss}) is the feature extracted from the pre-trained VGG-19.

3.4.5 Training loss

The overall training loss is formulated as:

$$\begin{aligned}\mathcal{L} = {} & \lambda_{\text{emd}}\mathcal{L}_{\text{emd}} + \lambda_{\text{sty}}\mathcal{L}_{\text{sty}} + \lambda_{\text{self-sim}}\mathcal{L}_{\text{self-sim}} \\ & + \lambda_{\text{cont}}\mathcal{L}_{\text{cont}} + \lambda_{\text{ide1}}\mathcal{L}_{\text{ide1}} + \lambda_{\text{ide2}}\mathcal{L}_{\text{ide2}} + \lambda_{\text{illum}}\mathcal{L}_{\text{illum}}\end{aligned} \tag{14}$$

where λ_* are weights to balance each term. Since we bulid our model on the MCCNet [9], the weights λ_{sty}, λ_{cont}, λ_{ide1}, λ_{ide2}, and λ_{illum} are set to 3, 10, 70, 1, and 3000 according to their implementation. And we set λ_{emd}, $\lambda_{\text{self-sim}}$ to 10, 16 following [27].

4 Experiment

4.1 Implementation details

We trained the network with MS-COCO [29] as the content images and WikiArt [30] as the style images. s Both datasets contain roughly 80,000 training images. To perform training the images are first rescaled to 300×300 sizes and then randomly cropped to 256×256 sizes for augmentation. At inference, our network can handle images of arbitrary size because it is fully convolutional. We used the Adam optimizer with a learning rate of 1e-4 and a batch size of 8 images. The training process lasts for 160K iterations on two Nvidia GTX 1080 Ti GPUs, where the encoder is fixed all time.

4.2 Comparing with previous methods

To evaluate our method, we compare it with other state-of-the-art arbitrary style transfer methods, including AdaIN [13], WCT [14], LST [15], SANet [16], MCCNet [9], and AdaAttN [18].

4.2.1 Image style transfer

The qualitative comparison are shown in Fig. 4, notice that none of the test images were observed during the training.

AdaIN [13] simply adjusts the mean and variance of the content feature, resulting in less appealing stylized results (1st and 3rd rows). WCT [14] generates much more vivid stylized images by modeling the the second-order statistics. However, it tends to produce a excessive stylization rendering a whole picture chaotic and showing many extraneous textures. LST [15] learns a transformation matrix to simulate the whitening and colorization in WCT and achieves the best speed performance, but the contour of objects has been distorted (face in the 3rd row). SANet [16] adopts the self-attention mechanism to capture local style patterns, but the content structure suffers severe damage (1st, 3rd and 4th rows). As mentioned above, it ignores the correlation between feature channels, so the model is not robust enough to tiny variations. AdaAttN [18] adaptively performs attentive normalization on per-point basis for feature distribution alignment. Their results consist of rich details and appropriate local style patterns, which achieve the best balance between style transfer and content preservation among all methods (1st and 4th rows). However, the stylized image seems totally unreasonable in some cases (color tone in 3rd row). MCCNet [9] designs the multi-channel attention module, producing relative clean results. Compared with it, our model has a better performance for introducing the rEMD loss and content-enhanced components (1st and 2nd rows).

4.2.2 Video style transfer

Qualitative As shown in Fig. 5, we use heat maps of difference to visualize the coherence between adjacent frames. The the heat map generated by ACNet is the closest to the original input, indicating that it has the best temporal stability. AdaIN

Fig. 4 Qualitative comparison with state-of-the-art methods on image style transfer

[13], WCT [14] and SANet [16] introduce a lot of flicking noise, and the semantic structure of the image has been seriously damaged. LST [15], MCCNet [9] and AdaAttN [18] are much stable, but there is still a gap with our method, proving the contributions of content-enhanced components to temporal stability.

Quantitative Following previous work, we conduct quantitative evaluation on the training set of MPI-Sintel [31] and DAVIS-2017 [32]. The MPI-Sintel dataset is initially for the evaluation of optical flow. It contains 35 long sequences, including motion blur, specular reflections, and other challenging cases. The DAVIS-2017 dataset is initially collected from the real world for video object segmentation. It contains 90 sequences with various objects of different motion types. For each method, we generate stylized frames on five styles.

We adopt the widely used flow warping error (FWE) to evaluate the temporal consistency:

$$\mathrm{FWE}\left(O_t, O_{t-1}\right) = M \odot \left\| O_t - \mathcal{W}^{f}_{t-1}\left(O_{t-1}\right) \right\|_2^2 \tag{15}$$

where $\mathcal{W}^{f}_{t-1}$ denotes the warping operation based on the backward optical flow between stylized output O_t and O_{t-1}, $M \in \{0, 1\}$ is the mask indicating areas where the optical flow is consistent and estimated with high confidence. We use the method in [3] to compute occlusion mask M_o and motion boundary mask M_m, and M can be obtained by:

$$M = 1 - \left(M_o \vee M_m\right) \tag{16}$$

where $\vee$ denotes logical operator OR.

Besides, as stated in [28], the temporal difference error (TDE) can indicate the ability of algorithms to preserve content affinity during stylization. Thus, we also employ it as an auxiliary metric:

$$\mathrm{TDE}\left(O_t, O_{t-1}\right) = \left\| \left(O_t - O_{t-1}\right) - \left(I_t - I_{t-1}\right) \right\|_2^2 \tag{17}$$

We calculate the average metric on each sequence and then average all the sequences to get the final metric. The evaluation results are shown in Tables 1 and 2. On the MPI-Sintel dataset, our method achieves the best FWE and TDE for all five styles. Compared with the previous state-of-the-art video style transfer method MCCNet, our method reduces 39.8% average FWE and 13.6% average TDE, respectively, greatly improves the stability. Similar results can be observed on the DAVIS-2017, where we can reduce 44.8% FWE and 13.8% TDE compared to MCCNet. These demonstrate that our method can generate coherent stylized frames with the least flicking artifacts.

4.2.3 Efficiency

We compare the efficiency of our method and other methods at three image resolutions: 256, 512 and 1024 pixels.

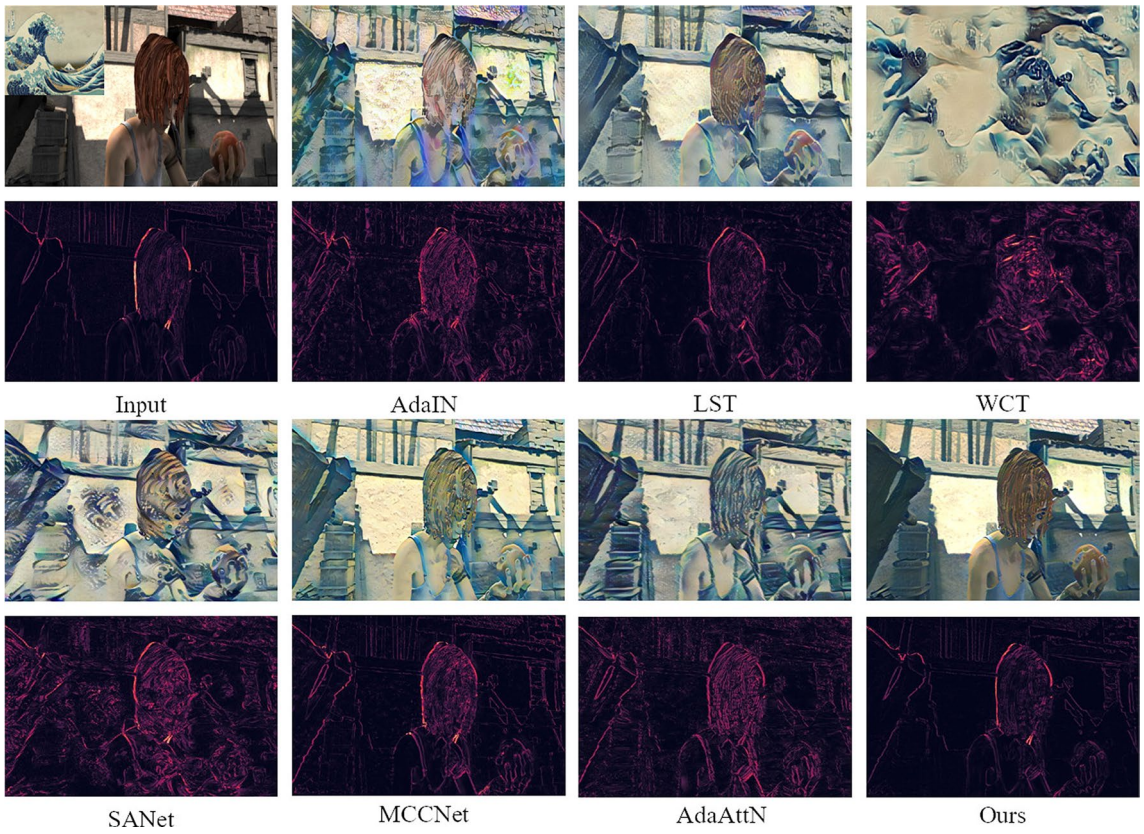

Fig. 5 Qualitative comparison on video style transfer. The first row shows the stylized frames. The second row shows the heat maps of the difference between adjacent frames

Table 1 Quantitative evaluation on the training set of MPI-Sintel ($\times 10^{-2}$)

Metric	Method	Style					Average
		Candy	Asheville	Sketch	Udnie	Wave	
FWE↓	AdaIN [13]	0.710	1.370	0.305	0.261	0.602	0.674
	LST [15]	0.408	1.155	0.258	0.172	0.411	0.481
	SANet [16]	0.926	1.861	0.448	0.464	0.827	0.905
	MCCNet [9]	0.316	0.859	0.195	0.125	0.348	0.369
	AdaAttN [18]	0.618	1.023	0.396	0.320	0.608	0.593
	Ours	**0.234**	**0.475**	**0.125**	**0.098**	**0.177**	**0.222**
TDE↓	AdaIN [13]	0.126	0.130	0.115	0.109	0.123	0.121
	LST [15]	0.115	0.125	0.107	0.098	0.108	0.111
	SANet [16]	0.140	0.139	0.128	0.128	0.132	0.133
	MCCNet [9]	0.108	0.115	0.097	0.092	0.101	0.103
	AdaAttN [18]	0.128	0.133	0.125	0.114	0.127	0.125
	Ours	**0.098**	**0.109**	**0.088**	**0.073**	**0.081**	**0.089**

The best result are marked in bold, and we ignore WCT [14] for its poor performance

All experiments are conducted using a single GTX 1080Ti under the same conditions. The inference time per frame (in second) is presented in Table 3. LST [15] achieves the best run time performance due to its simple linear transformation. Our method is just slightly behind LST [15],

Table 2 Quantitative evaluation on the training set of DAVIS-2017 ($\times 10^{-2}$)

Metric	Method	Style					Average
		Candy	Asheville	Sketch	Udnie	Wave	
FWE↓	AdaIN [13]	0.686	1.610	0.377	0.293	0.630	0.719
	LST [15]	0.482	1.425	0.320	0.209	0.494	0.586
	SANet [16]	0.984	2.162	0.510	0.526	0.931	1.023
	MCCNet [9]	0.380	1.112	0.234	0.150	0.432	0.462
	AdaAttN [18]	0.720	1.153	0.458	0.377	0.655	0.673
	Ours	**0.268**	**0.536**	**0.144**	**0.119**	**0.209**	**0.255**
TDE↓	AdaIN [13]	0.130	0.137	0.124	0.117	0.126	0.127
	LST [15]	0.121	0.131	0.114	0.105	0.113	0.117
	SANet [16]	0.144	0.145	0.132	0.132	0.133	0.137
	MCCNet [9]	0.116	0.122	0.104	0.098	0.105	0.109
	AdaAttN [18]	0.130	0.135	0.127	0.118	0.127	0.127
	Ours	**0.103**	**0.115**	**0.089**	**0.080**	**0.085**	**0.094**

The best result are marked in bold, and we ignore WCT [14] for its poor performance

Table 3 Efficiency comparison (in second) under multiple resolutions on a single GTX 1080Ti

Method	Resolution		
	256	512	1024
AdaIN [13]	0.013	0.044	0.163
WCT [14]	0.986	1.237	4.052
LST [15]	0.008	0.031	0.114
SANet [16]	0.018	0.065	0.306
MCCNet [9]	0.009	0.032	0.116
AdaAttN [18]	0.023	0.088	0.486
Ours	0.009	0.033	0.133

but the stylization quality and temporal stability are much better.

4.3 Ablation study

In this section, we conduct the ablation experiments to demonstrate the effectiveness of the introduced rEMD loss, AdaIN-based skip connections and self-similarity loss. Since we build our model based on the MCC module [9], MCCNet is employed as the baseline method.

Qualitative The qualitative results are shown in Fig. 6. The rEMD loss helps the model to focus on the main characteristics of style images, therefore the textures and strokes of output images are closer to the style input. Besides, it can remove the strange color blocks (face in the 1st row). The self-similarity loss maintains the relative relationship between semantic elements, and the images become much cleaner (3rd row). AdaIN-based skip connections introduced rich stylized details, but may interfere with aesthetics. When combining the two content-enhanced components, the stylization strength would be further reduced. For example, in the 4th row, strokes of the sky region would not be similar to the style image. And the strange color blots appear again (1st row). By integrating these three ingredients, ACNet achieves the best stylization quality.

Quantitative As shown in Tables 4 and 5, all the three modules can improve the FWE and TDE metric, especially self-similarity loss and AdaIN-based skip connections. The self-similarity loss helps to maintain the relative relationship between semantic elements in the image, so the stylized results of adjacent frames would not change violently to hurt such relationship. As for AdaIN-based skip connections, it brings more stylized details into the decoder. Because the MCC module is performed on feature maps extracted from a deep layer (ReLUx1), the decoder is responsible for restoring spatial details. If there is no additional information, the uncertainty of the recovery process would be greater, resulting in unstable textures in the stylized image. Thus AdaIN-based skip connections can improve temporal consistency by reducing the uncertainty during stylization. Averaged over the two datasets, the self-similarity loss reduces 27.7% FWE and 10.2% TDE, and the AdaIN-based skip connections reduces 40.2% FWE and 9.6% TDE. Even though rEMD loss is designed for better stylization quality, it can also improve both metrics (5.8% FWE and 5.5% TDE) slightly due to capturing the main style pattern. However, the best performance is achieved by only employing content-enhanced components (AdaIN-based skip connections and self-similarity loss). We argue that it is due to the trade-off between stylization strength and temporal consistency, two conflicting goals in essence. Exploring a suitable combination of the weight of each loss function to control this balance better is a significant part in our future work. These results demonstrate again that enhancing the content structure leads to better temporal consistency.

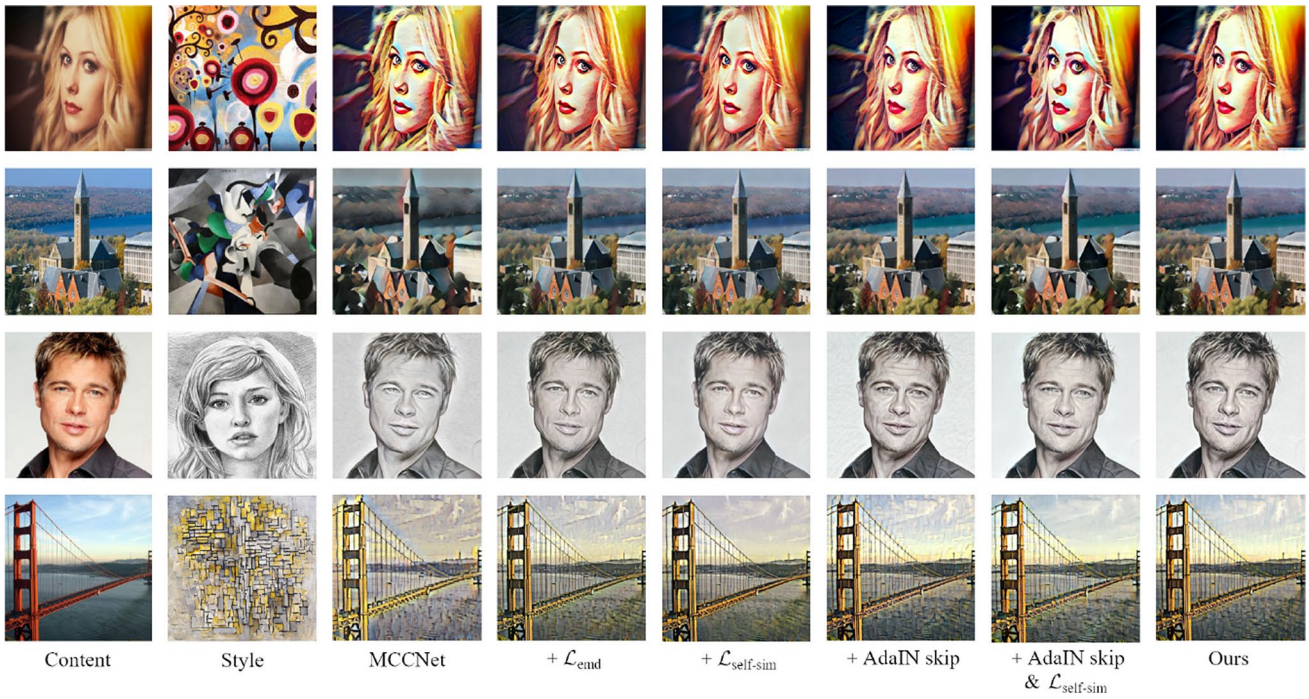

Fig. 6 Qualitative comparison of the ablation experiments on image style transfer

Table 4 Quantitative evaluation of ablation experiments on the training set of MPI-Sintel ($\times 10^{-2}$)

Metric	Method	Style					Average
		Candy	Asheville	Sketch	Udnie	Wave	
FWE↓	Baseline	0.316	0.859	0.195	0.125	0.348	0.369
	$+\mathcal{L}_{rEMD}$	0.237	0.611	0.159	0.101	0.257	0.273
	$+\mathcal{L}_{self-sim}$	**0.230**	0.597	0.135	0.103	0.271	0.267
	+ AdaIN skip	0.247	0.492	0.131	0.099	0.179	0.230
	+ content-enhanced	0.236	**0.461**	**0.116**	**0.097**	**0.171**	**0.216**
	Full model	0.234	0.475	0.125	0.098	0.177	0.222
TDE↓	Baseline	0.108	0.115	0.097	0.092	0.101	0.103
	$+\mathcal{L}_{rEMD}$	0.101	0.113	0.094	0.083	0.092	0.097
	$+\mathcal{L}_{self-sim}$	0.097	0.110	0.087	0.078	0.090	0.092
	+ AdaIN skip	0.102	0.112	0.092	0.077	0.084	0.093
	+ content-enhanced	**0.096**	**0.107**	**0.083**	**0.070**	**0.077**	**0.087**
	Full model	0.098	0.109	0.088	0.073	0.081	0.089

The bold results represents the best ones in each part

5 Conclusion

In this paper, we propose a new video style transfer framework without the dependency on the inefficient optical flow. In our model, we adopt multi-channel correlation module as basic style transfer module which considers the inter-channel relationship of features, and use Earth Movers' Distance to further improve the stylization quality. We also combine AdaIN-based skip connections and self-similarity loss to maintain the semantic structure during stylization. We demonstrate the effectiveness of model, which can produce consistent results with reasonable style patterns and clear content structure in real-time.

Table 5 Quantitative evaluation of ablation experiments on the training set of DAVIS-2017 ($\times 10^{-2}$)

Metric	Method	Style					Average
		Candy	Asheville	Sketch	Udnie	Wave	
FWE↓	Baseline	0.380	1.112	0.234	0.150	0.432	0.462
	$+\mathcal{L}_{r\text{EMD}}$	0.294	0.779	0.204	0.133	0.341	0.350
	$+\mathcal{L}_{\text{self-sim}}$	0.279	0.746	0.172	0.134	0.337	0.334
	+ AdaIN skip	0.282	0.559	0.149	0.122	0.212	0.265
	+ content-enhanced	**0.265**	**0.518**	**0.133**	**0.117**	**0.200**	**0.247**
	Full model	0.268	0.536	0.144	0.119	0.209	0.255
TDE↓	Baseline	0.116	0.122	0.104	0.098	0.105	0.109
	$+\mathcal{L}_{r\text{EMD}}$	0.108	0.120	0.099	0.091	0.099	0.103
	$+\mathcal{L}_{\text{self-sim}}$	0.104	0.115	0.093	0.088	0.096	0.099
	+ AdaIN skip	0.107	0.117	0.094	0.085	0.089	0.098
	+ content-enhanced	**0.099**	**0.112**	**0.084**	**0.077**	**0.080**	**0.090**
	Full model	0.103	0.115	0.089	0.080	0.085	0.094

The bold results represents the best ones in each part

Table 6 Details of the encoder

Layer	Kernel (Size, Numbers)	Stride	Output size (Dimensions × Height × Weight)
Input	–	–	$3 \times H \times W$
Conv_in	$1 \times 1, 3$	1	$3 \times H \times W$
Conv1_1*	$3 \times 3, 64$	1	$64 \times H \times W$
Conv1_2	$3 \times 3, 64$	1	$64 \times H \times W$
Maxpool	2×2	2	$64 \times \frac{H}{2} \times \frac{W}{2}$
Conv2_1*	$3 \times 3, 128$	1	$128 \times \frac{H}{2} \times \frac{W}{2}$
Conv2_2	$3 \times 3, 128$	1	$128 \times \frac{H}{2} \times \frac{W}{2}$
Maxpool	2×2	2	$128 \times \frac{H}{4} \times \frac{W}{4}$
Conv3_1*	$3 \times 3, 256$	1	$256 \times \frac{H}{4} \times \frac{W}{4}$
Conv3_2	$3 \times 3, 256$	1	$256 \times \frac{H}{4} \times \frac{W}{4}$
Conv3_3	$3 \times 3, 256$	1	$256 \times \frac{H}{4} \times \frac{W}{4}$
Conv3_4	$3 \times 3, 256$	1	$256 \times \frac{H}{4} \times \frac{W}{4}$
Maxpool	2×2	2	$256 \times \frac{H}{8} \times \frac{W}{8}$
Conv4_1	$3 \times 3, 512$	1	$512 \times \frac{H}{8} \times \frac{W}{8}$

Input are RGB images

* indicates there exist skip connections in this layer

Table 7 Details of the decoder

Layer	Kernel (Size, Numbers)	Stride	Output Size (Dimensions × Height × Weight)
Input	–	–	$512 \times \frac{H}{8} \times \frac{W}{8}$
Resblock1	$\begin{pmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{pmatrix}$	1	$512 \times \frac{H}{8} \times \frac{W}{8}$
		1	$512 \times \frac{H}{8} \times \frac{W}{8}$
Conv1	$3 \times 3, 256$	1	$256 \times \frac{H}{8} \times \frac{W}{8}$
Upsample	–	–	$256 \times \frac{H}{4} \times \frac{W}{4}$
Resblock2*	$\begin{pmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{pmatrix}$	1	$256 \times \frac{H}{4} \times \frac{W}{4}$
		1	$256 \times \frac{H}{4} \times \frac{W}{4}$
Conv2	$3 \times 3, 128$	1	$128 \times \frac{H}{4} \times \frac{W}{4}$
Upsample	–	–	$128 \times \frac{H}{2} \times \frac{W}{2}$
Conv3_1*	$3 \times 3, 128$	1	$128 \times \frac{H}{2} \times \frac{W}{2}$
Conv3_2	$3 \times 3, 128$	1	$64 \times \frac{H}{2} \times \frac{W}{2}$
Upsample	-	–	$64 \times H \times W$
Conv4_1*	$3 \times 3, 64$	1	$64 \times H \times W$
Conv4_2	$3 \times 3, 64$	1	$3 \times H \times W$

Input are stylized features, and we adopt the bilinear upsample to recover the spatial size of feature maps

* indicates there exist skip connections in this layer

Appendix A Details of encoder and decoder

As stated in the main paper, we employ the pre-trained VGG-19 as the encoder and constitute the decoder with residual blocks. Both networks consists of small size convolutional blocks, and each convolutional layer is followed by a ReLU layer (nonlinear activation function). Tables 6 and 7 provides full details of the encoder and decoder.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10044-022-01106-y.

Author Contributions HL: Conceptualization, Methodology, Investigation, Validation, Writing-original draft, Writing-review and editing. MW: Conceptualization, Methodology, Investigation, Validation, Writing-review and editing. YL: Conceptualization, Validation, Project administration, Writingreview and editing. JK: Conceptualization, Investigation, Validation, Project administration

Funding This work was supported by Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Gatys LA, Ecker AS, Bethge M (2016) Image style transfer using convolutional neural networks. In: IEEE conference on Computer Vision and Pattern Recognition, pp. 2414–2423
2. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
3. Ruder M, Dosovitskiy A, Brox T (2016) Artistic style transfer for videos. In: German Conference on Pattern Recognition, pp. 26–36. Springer
4. Huang H, Wang H, Luo W, Ma L, Jiang W, Zhu X, Li Z, Liu W (2017) Real-time neural style transfer for videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 783–791
5. Gupta A, Johnson J, Alahi A, Fei-Fei L (2017) Characterizing and improving stability in neural style transfer. In: IEEE International Conference on Computer Vision, pp. 4067–4076
6. Chen X, Zhang Y, Wang Y, Shu H, Xu C, Xu C (2020) Optical flow distillation: towards efficient and stable video style transfer. In: European Conference on Computer Vision, pp. 614–630. Springer
7. Chen D, Liao J, Yuan L, Yu N, Hua G (2017) Coherent online video style transfer. In: IEEE International Conference on Computer Vision, pp. 1105–1114
8. Wang W, Yang S, Xu J, Liu J (2020) Consistent video style transfer via relaxation and regularization. IEEE Trans Image Process 29:9125–9139
9. Deng Y, Tang F, Dong W, Huang H, Ma C, Xu C (2020) Arbitrary video style transfer via multi-channel correlation. arXiv preprint arXiv:2009.08003
10. Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, pp. 694–711. Springer
11. Chen D, Yuan L, Liao J, Yu N, Hua G (2017) Stylebank: an explicit representation for neural image style transfer. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1897–1906
12. Dumoulin V, Shlens J, Kudlur M (2016) A learned representation for artistic style. arXiv preprint arXiv:1610.07629
13. Huang X, Belongie S (2017) Arbitrary style transfer in real-time with adaptive instance normalization. In: IEEE International Conference on Computer Vision, pp. 1501–1510
14. Li Y, Fang C, Yang J, Wang Z, Lu X, Yang M-H (2017) Universal style transfer via feature transforms. arXiv preprint arXiv:1705.08086
15. Li X, Liu S, Kautz J, Yang M-H (2019) Learning linear transformations for fast image and video style transfer. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3809–3817
16. Park DY, Lee KH (2019) Arbitrary style transfer with style-attentional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5880–5888
17. Yao Y, Ren J, Xie X, Liu W, Liu Y-J, Wang J (2019) Attention-aware multi-stroke style transfer. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1467–1475
18. Liu S, Lin T, He D, Li F, Wang M, Li X, Sun Z, Li Q, Ding E (2021) Adaattn: revisit attention mechanism in arbitrary neural style transfer. In: IEEE International Conference on Computer Vision, pp. 6649–6658
19. Lai W-S, Huang J-B, Wang O, Shechtman E, Yumer E, Yang M-H (2018) Learning blind video temporal consistency. In: European Conference on Computer Vision, pp. 170–185
20. Gao W, Li Y, Yin Y, Yang M-H (2020) Fast video multi-style transfer. In: IEEE Winter Conference on Applications of Computer Vision, pp. 3222–3230
21. Wang T-C, Liu M-Y, Zhu J-Y, Liu G, Tao A, Kautz J, Catanzaro, B. (2018) Video-to-video synthesis. arXiv preprint arXiv:1808.06601
22. Zhou Y, Xu X, Shen F, Gao L, Lu H, Shen, HT (2020) Temporal denoising mask synthesis network for learning blind video temporal consistency. In: ACM International Conference on Multimedia, pp. 475–483
23. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778
24. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008
25. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223. PMLR
26. Li Y, Wang N, Liu J, Hou X (2017) Demystifying neural style transfer. arXiv preprint arXiv:1701.01036
27. Kolkin N, Salavon J, Shakhnarovich G. (2019) Style transfer by relaxed optimal transport and self-similarity. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 10051–10060
28. Xu K, Wen L, Li G, Qi H, Bo L, Huang Q (2021) Learning self-supervised space-time cnn for fast video style transfer. IEEE Trans Image Process 30:2501–2512
29. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European Conference on Computer Vision, pp. 740–755. Springer
30. Saleh B, Elgammal A (2015) Large-scale classification of fine-art paintings: learning the right metric on the right feature. arXiv preprint arXiv:1505.00855
31. Butler DJ, Wulff J, Stanley GB, Black MJ (2012) A naturalistic open source movie for optical flow evaluation. In: European Conference on Computer Vision, pp. 611–625. Springer

32. Pont-Tuset J, Perazzi F, Caelles S, Arbeláez P, Sorkine-Hornung A, Gool LV (2018) The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675