

Exploring the Temporal Consistency of Arbitrary Style Transfer: A Channelwise Perspective

Xiaoyu Kong, Yingying Deng, Fan Tang[✉], Weiming Dong[✉], *Member, IEEE*, Chongyang Ma, *Member, IEEE*, Yongyong Chen, Zhenyu He[✉], *Senior Member, IEEE*, and Changsheng Xu, *Fellow, IEEE*

Abstract—Arbitrary image stylization by neural networks has become a popular topic, and video stylization is attracting more attention as an extension of image stylization. However, when image stylization methods are applied to videos, unsatisfactory results that suffer from severe flickering effects appear. In this article, we conducted a detailed and comprehensive analysis of the cause of such flickering effects. Systematic comparisons among typical neural style transfer approaches show that the feature migration modules for state-of-the-art (SOTA) learning systems are ill-conditioned and could lead to a channelwise misalignment between the input content representations and the generated frames. Unlike traditional methods that relieve the misalignment via additional optical flow constraints or regularization modules, we focus on keeping the temporal consistency by aligning each output frame with the input frame. To this end, we propose a simple yet efficient multichannel correlation network (MCCNet), to ensure that output frames are directly aligned with inputs in the hidden feature space while maintaining the desired style patterns. An inner channel similarity loss is adopted to eliminate side effects caused by the absence of nonlinear operations such as softmax for strict alignment. Furthermore, to improve the performance of MCCNet under complex light conditions, we introduce an illumination loss during training. Qualitative and quantitative evaluations demonstrate that MCCNet performs well in arbitrary video and image style transfer tasks. Code is available at <https://github.com/kongxiuxiu/MCCNetV2>.

Index Terms—Arbitrary stylization, channel correlation, cross-domain, feature migration.

Manuscript received 10 November 2021; revised 15 July 2022 and 28 November 2022; accepted 12 December 2022. Date of publication 6 January 2023; date of current version 4 June 2024. This work was supported by the National Key R&D Program of China under Grant 2020AAA0106200; in part by the National Science Foundation of China under Grant U20B2070, Grant 61832016, Grant 62102162, Grant 62172126, and Grant 62106063; in part by the Major Key Project of Peng Cheng Laboratory under Grant PCL2021A03-1; in part by the Shenzhen Research Council under Grant JCYJ20210324120202006; and in part by the Guangdong Natural Science Foundation under Grant 2022A1515010819. (Xiaoyu Kong and Yingying Deng contributed equally to this work.) (Corresponding author: Fan Tang.)

Xiaoyu Kong is with the School of Artificial Intelligence, Jilin University, Changchun 130012, China, and also with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518073, China.

Yingying Deng, Weiming Dong, and Changsheng Xu are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Fan Tang is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: tfan.108@gmail.com).

Chongyang Ma is with the Kuaishou Technology, Beijing 100085, China.

Yongyong Chen is with the Department of Computer Science, Harbin Institute of Technology, Shenzhen 518073, China.

Zhenyu He is with the Department of Computer Science, Harbin Institute of Technology, Shenzhen 518073, China, and also with the Peng Cheng Laboratory, Shenzhen 518055, China.

This article has supplementary downloadable material available at <https://doi.org/10.1109/TNNLS.2022.3230084>, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2022.3230084

I. INTRODUCTION

ARBITRARY style transfer (AST) for natural images or videos is an attractive research direction that successfully bridges artificial intelligence techniques with art and has become a trending topic in industrial applications [1], [2], [3], [4]. Given a content image and a painting image, a desired image style transfer method can render the content image into the artistic style referenced by the painting image. Deep convolutional neural networks (CNNs) have been widely studied for artistic image generation and translation [3], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. As the extension of image stylization, video style transfer is a more challenging task that needs to generate good stylization results per frame and consider the continuity between adjacent frames of the video. Although the existing image stylization methods can generate satisfactory results for still images, they fail to handle the consecutive frames in a video suitably. As shown in Fig. 1, when slight changes happen in two adjacent frames, the stylized frames differ considerably; the frames in a video are stylized, and the generated video looks inconsecutive, which is called the flickering effect.

Methods [18], [19], [20] for keeping the temporal consistency of the generated videos can be divided into the *sequence-based* and *frame-based* approaches. The sequence-based video stylization methods measure the interframe correlation of input videos by additional information, such as optical flow, to keep the temporal consistency in the results [21], [22], [23]. Gao et al. [24] proposed a multistyle video transfer model that estimates the light flow and introduces temporal constraint. However, the sequence-based approaches highly depend on the accuracy of optical flow calculation. Adding optical flow to constrain the coherence of stylized videos makes the network difficult to train when applied to an AST model. The frame-based video style transfer methods do not measure the interframe correlation of input videos. Additional constraints [25] or extra stable modules [26] are applied to the generated videos to maintain the continuity between adjacent frames, which may impair the stylized effects.

The sequence-based or frame-based video stylization approaches focus on improving the temporal consistency by additional conditions or constraints. However, for video stylization, the input style reference is fixed and the content video usually does not contain the flickering effects [Fig. 1(a)]. The cause of flickering in the generated videos is worth exploring. Following the encoder–transform–decoder pipeline, the image

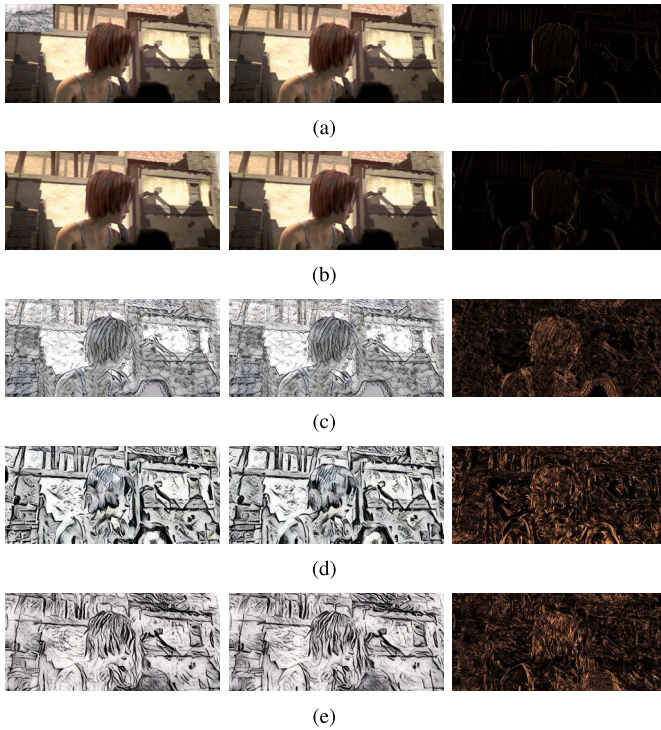


Fig. 1. Visual consistency of (a) input video frames, (b) reconstructions by AE, (c) stylized results by AdaIN, (d) stylized results by WCT, and (e) stylized results by SANet (a) Two adjacent frames of an input video *Sintel*. (b) Reconstruction frames by a pretrained AE. (c) Stylized frameworks by AdaIN [15]. (d) Stylized frameworks by WCT [16]. (e) Stylized frameworks by SANet [17].

AST approaches can be viewed as an autoencoder (AE) [27] with a cross-domain feature migration. When a pretrained autoencoder in the encoder–decoder architecture is used to reconstruct the two frames, a similar residual map shows that the encoder–decoder successfully maintains the consecutiveness of the input frames [Fig. 1(b)]. However, when the same two adjacent frames are stylized by the adaptive instance normalization (AdaIN) [15], whitening and coloring transformation (WCT) [16], and style-attention network (SANet) [17], the noisy residual maps show that the consecutiveness of the inputs is broken [Fig. 1(c)–(e)]. Such observation motivates us to determine the cause of the flickering effects of frame-based video style transfer.

In this work, we first revisit the feature migration of the existing style transfer methods and point out the advantage of strictly aligned feature migration in maintaining the consistency of videos. Then, we propose a frame-based multichannel correlation network (MCCNet) for temporally coherent video style transfer, which has a satisfactory performance in video style transfer without evolving the calculation of optical flow or any consistency constraints. MCCNet adaptively rearranges style representations on the basis of content representations by considering the multichannel correlation of these representations, thus making style patterns suitable for content structures. By further merging the rearranged style and content representations, we generate features that can be decoded into stylized results with clear content structures and vivid style patterns. MCCNet aligns the generated features to content

features; thus, slight changes in adjacent frames will not cause flickering in the stylized video. The inner channel similarity loss is added to enhance the correlation within each channel to keep the consistency of the style pattern. Furthermore, the illumination variation among consecutive frames influences the stability of video style transfer. Thus, we add random Gaussian noise to the content image to simulate illumination variations and propose an illumination loss to stabilize the generated videos and avoid flickering. Our method is suitable for stable video style transfer and can generate single-image style transfer results with well-preserved content structures and vivid style patterns.

In summary, our main contributions are as follows.

- 1) We comprehensively analyze typical encoder–transform–decoder-based neural style transfer (NST) methods and point out the advantage of feature-aligned migration in maintaining the temporal consistency of video stylization.
- 2) We propose MCCNet for framed-based video style transfer by strictly aligning cross-domain features with input videos to render coherent results. An inner channel similarity loss is further proposed to relieve the side visual effect by strict feature alignment.
- 3) We propose an illumination loss to make the style transfer process increasingly stable so that our model can be flexibly applied to videos with complex light conditions.
- 4) Numerous experiments and ablation studies verify the importance of feature-aligned migration and demonstrate the effectiveness of our method on image and video stylization tasks.

This article is an extension of our earlier publication [28]. The changes can be summarized into three aspects: principles supplementing, method improvement, and experiment enrichment. First, this article analyzes the feature migration of the existing AST methods from the perspective of channel interaction in Section III. Second, the original MCCNet focuses on the correlation among channels and reaches the strict alignment at the cost of abandoning the softmax operation, which would cause severe highlight effects around the object. We propose an inner channel similarity loss to enhance the correlation inner each channel and eliminate the highlight effect caused by the old model. Third, we conduct more experiments and related analyses. We discuss the function of aligned feature migration by proposing an AdaIN-variation and show the stability brought by feature-aligned migration. Ablation studies explore the separate effects of our proposed pipeline.

II. RELATED WORK

A. Image Style Transfer

Image style transfer has been widely studied in recent years [5], [6], [29], [30], [31]. Essentially, it enables the generation of artistic paintings without the expertise of a professional painter. Gatys et al. [5] found that the inner products of the feature maps in CNNs can be used to

represent style and proposed an NST method through continuous optimization iterations. However, the optimization process is time-consuming and cannot be widely used. Dumoulin et al. [32] proposed conditional instance normalization (CIN), which allows learning multiple styles within one model by reducing a style image into a point in the embedded space. Several methods achieve AST by aligning the second-order statistics of style and content images [15], [16], [33]. Huang and Serge [15] proposed an AST method by adopting AdaIN, which normalizes content features using the mean and variance of style features. Li et al. [16] used WCT to render content images with style patterns. Wang et al. [33] adopted deep feature perturbation in a WCT-based model to achieve diversified AST. However, these holistic transformations lead to unsatisfactory results. Park and Lee [17] proposed a SANet to obtain abundant style patterns in the generated results but failed to maintain distinct content structures. Yao et al. [34] proposed an attention-aware multistroke style transfer (AAMS) model by adopting self-attention to a style swap-based image transfer method, which highly relies on the accuracy of the attention map used. Deng et al. [35] proposed a multiadaptation style transfer (MAST) method to disentangle content and style features and combine them adaptively. However, some results are rendered with uncontrollable style patterns. Meanwhile, Wei et al. [36] further analyzed the influence of different neural network designs on style transfer tasks and pointed out that the Transformer-based models cannot render style patterns properly due to its shape bias.

Although these methods perform well on images, they lead to flickering effects for frame-based video style transfer without considering temporal consistency constraints. In this article, we focus on the network design of transfer module and find that even without temporal consistency, image style transfer can avoid flickering via the feature alignment between transformed and content features.

B. Video Style Transfer

Most video style transfer methods rely on the existing image style transfer methods [21], [22], [24]. Ruder et al. [21] added a temporal constraint to NST [5] to avoid flickering. However, the optimization-based method is inefficient for video style transfer. Chen et al. [22] proposed a feed-forward network for fast video style transfer by incorporating temporal information. Chen et al. [37] distilled knowledge from the video style transfer network with the optical flow to a student network to avoid optical flow estimation in the test stage. Gao et al. [24] adopted CIN for multistyle video transfer, and the approach incorporated one FlowNet and two convolutional long short-term memory (ConvLSTM) modules to estimate light flow and introduce a temporal constraint. The effect of the sequence-based methods highly depended on the accuracy of optical flow calculation. Using inappropriate optical flows will cause ghosting artifacts. Moreover, estimating optical flows is computationally expensive, which makes processing high-resolution or long videos difficult. Wang et al. [38] proposed a novel interpretation of temporal consistency without optical flow estimation for efficient zero-shot style transfer.

Li et al. [26] learned a transformation matrix for AST. They found that the normalized affinity for generated features is the same as that for content features and is thus suitable for frame-based video style transfer. However, the style patterns in their stylized results are not noticeable.

To avoid using optical flow estimation while maintaining video continuity, we aim to design an alignment transform operation to achieve stable video style transfer with vivid style patterns. Without additional conditions or constraints for temporal consistency, the proposed MCCNet is more efficient and does not need to balance between temporal consistency and stylization effect.

C. Channel Interaction

As deep neural networks extract and fuse channelwise information, the interaction among channels is an important topic and has been investigated in recent years. Hu et al. [39] proposed Squeeze-and-Extraction Network (SENet) to explicitly model the interdependencies between the channels. Woo et al. [40] used a channelwise attention mask to push a network focus on important features and suppress unnecessary ones. The channel interaction helps improve the representational power of networks, thus achieving significant improvements in object recognition. In addition to sorting tasks, Fu et al. [41], [42] introduced channel interaction to the segmentation task via a channel attention module to selectively emphasize interdependent channel maps by integrating associated features among all the channel maps. Ji et al. [43] designed the self- and cross-attention modules to preserve the saliency correlation and improve intraframe salient detection consistency in salient object detection tasks. The channel intersection improves the representational power of learning systems and performs well in various tasks; thus, importing channel intersection into the style transfer task is natural, and the strengthening of channel correlation could be favorable for simple and complex style pattern transfer.

III. TEMPORAL CONSISTENCY ANALYSIS

A. Overview

Most AST methods adopt an encoder–transform–decoder framework [Fig. 2(a)] based on an AE [27] architecture and a transfer module. The AE can extract the low-dimensional representation f from an input image I_{in} via the encoder $\mathbb{E}$ and then reconstruct f to the output image I_{out} with the decoder $\mathbb{D}$ which is trained to be the same with the input image I_{in} . The procedure can be formulated as follows:

$$\begin{aligned} f &= \mathbb{E}(I_{\text{in}}) \\ I_{\text{out}} &= \mathbb{D}(f) \\ \text{s.t. } I_{\text{out}} &= I_{\text{in}}. \end{aligned} \quad (1)$$

Research has explored the potential of such an encoder–decoder framework and successfully applied it to AST tasks. Given a content image I_c and a style image I_s , the corresponding feature maps are obtained by $f_c = \mathbb{E}(I_c)$ and $f_s = \mathbb{E}(I_s)$. Obtaining content representations f_c of the content image, extra style embedding f_s can be added to f_c to obtain the stylized representation f_{cs} , which maintains the

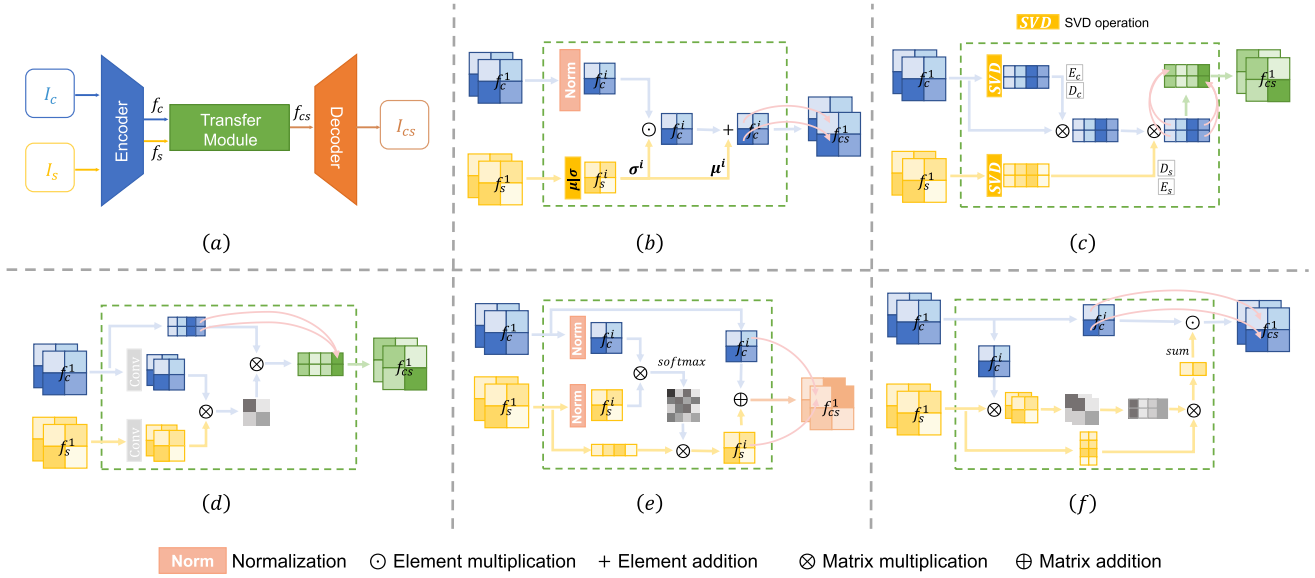


Fig. 2. Overall structure of different style transfer modules. The pink arrows indicate the alignment of features. (a) Encoder–transform–decoder. (b) AdaIN. (c) WCT. (d) Linear. (e) SANet. (f) MSSNet.

structure of the content image while being rendered to the desired style. The procedure of AST can be formulated as follows:

$$\begin{aligned}
 f_{cs} &= \mathbb{T}(f_c, f_s) \\
 I_{cs} &= \mathbb{D}(f_{cs}) \\
 \text{s.t. } I_{cs}^{\text{content}} &= I_c \\
 I_{cs}^{\text{style}} &= I_s
 \end{aligned} \quad (2)$$

where $\mathbb{T}(\cdot)$ can be any feature migration module.

A comparison between (1) and (2) shows that the main differences between AE and AST lie in the constraints and the feature migration module $\mathbb{T}(\cdot)$. By restricting each pixel of I_{out} to be the same as I_{in} , the AE could generate coherent outputs when applied to video reconstruction. However, AST tasks do not share the same strong constraint. Previous researchers focused on involving new constraints or regulations in (2) to ensure that the AST approaches could be applied to video stylization tasks but ignore the influence of $\mathbb{T}(\cdot)$. Inappropriate $\mathbb{T}(\cdot)$ will enlarge the slight changes between adjacent frames and raise flickering effects.

In this article, we point out that the key to extend image style transfer approaches to frame-based video style transfer is to maintain the consistency in the transfer module $\mathbb{T}(\cdot)$. We analyze the transfer module in different AST approaches in Section III-B. We focus on comparisons of different methods from the channelwise perspective. Then, we propose a prototype of feature-aligned migration as transfer module $\mathbb{T}(\cdot)$ which could not cause flickering effects for video stylization in Section III-C. In Section IV, we realize the prototype by multichannel correlation and propose the MCCNet.

B. Feature Migration of Existing Algorithm

To analyze the cause of flickering effects, we study the feature migration of different AST algorithms. Four typical

AST approaches are selected for comparison: AdaIN [15], WCT [16], SANet [17], and Linear [26]. AdaIN and WCT are the most representative AST methods that adopt the encoder–transform–decoder architecture. Linear relieves the flickering effect by preserving the dense pairwise relationships among pixels in the content image. SANet takes the attention mechanism into consideration which undergoes a different kind of feature alignment. The mechanism of different methods is shown in Fig. 2.

AdaIN is one of the most widely mentioned AST methods. As shown in Fig. 2(b), the transformation by AdaIN can be summarized as

$$f_{cs}^i = K f_c^i + \text{bias} \quad (3)$$

which means f_{cs} and f_c satisfy a linear relationship with a bias term at each channel separately, without any interchannel correlation. AdaIN only migrates the global statistics in each channel by which f_{cs}^i aligns with f_c^i . The neglect of local details and the lack of communication with other channels will reduce the stylization quality, especially when dealing with complicated style patterns.

Although AdaIN directly migrates the global statistics to f_c , which makes f_{cs} and f_c satisfy a linear relationship, the bias term at each channel negatively affects the consistency of f_{cs} . More discussions about the analysis of the temporal consistency of AdaIN are given in Section V-E.

WCT is based on pretrained AE and processes f_c and f_s via whitening and coloring operation. The transformation of WCT made by the transformation matrixes' multiplication can be written into one linear transformation matrix M

$$f_{cs} = M f_c. \quad (4)$$

f_{cs} obtained from a matrix multiplication generates new features of each pixel only with the pixels at the same location of multiple channels.

Comparing WCT with AdaIN from channelwise interaction is interesting; WCT merely focuses on the local style transfer using multiple channel information of f_s without global statistics into consideration, which is completely contrary to AdaIN. The different migration method causes WCT to perform poorly in generating global patterns. As shown in Fig. 2(c), WCT does not keep a linear relationship between f_c and f_{cs} like AdaIN does but satisfies a new relationship: linear transformation. However, the calculation of transformation matrix M breaks the consistency between input frames: f_{cs} cannot successfully hold the consistency within either inner channel or interchannel after the matrix multiplication operation.

Linear is similar to WCT because it proposes a learnable linear transformation matrix, which is conditioned on an arbitrary pair of content and style images to ensure that the stylized video has a similar affinity as the input video. The overall architecture is shown in Fig. 2(d). The transfer module of Linear can be described as (4), where M is a transfer matrix learned based on covariance matrices of content features and style features.

The main difference between WCT and Linear is the condition that the transformation matrix M should satisfy. Linear requires the transformation matrix M to satisfy the condition that the affinities of f_c and f_{cs} are equal to each other to keep the consistency. Unlike WCT, we could conclude that it is not the linear transformation matrix that preserves the consistency of output frames; instead, the additional constraints on the M make the output frames consistent. However, as shown in Section V-B, the visual effect of Linear contains fewer style patterns than other AST approaches.

SANet uses a self-attention mechanism in the transfer module to conduct style transfer with global and local patterns. As shown in Fig. 2(e), SANet calculates the attention between f_s^i and f_c^i and then applies the attention mask to f_s^i . The attention calculation takes place at each channel and is imposed to the corresponding f_s^i . The entanglement hurts the original continuity preserved by f_c , resulting in significant flickering on output videos.

C. Feature-Aligned Migration

For video style transfer, the input video is coherent with no flickering effects, which means that the input frames satisfy the following temporal consistency:

$$\|I_n - W_{I_m \rightarrow I_n}(I_m)\| < \delta \quad (5)$$

where I_n and I_m are the n th and m th frames of the input video, respectively, and $W_{I_m \rightarrow I_n}$ represents the corresponding optical flow that warps I_m to I_n .

As mentioned in Section III-A, basic operations (such as convolution and pooling in CNN), training target (losses), and the feature migration module $\mathbb{T}(\cdot)$ are the three components of the encoder–transformer–decoder-based AST approaches. Previous research focused on designing proper $\mathbb{T}(\cdot)$ for improving the stylization effect and used additional temporal information or losses for generating consistent results. In this article, we pay attention to the transform module $\mathbb{T}(\cdot)$. Given two adjacent frames I_n and I_{n+1} , we record the inputs of $\mathbb{T}(\cdot)$

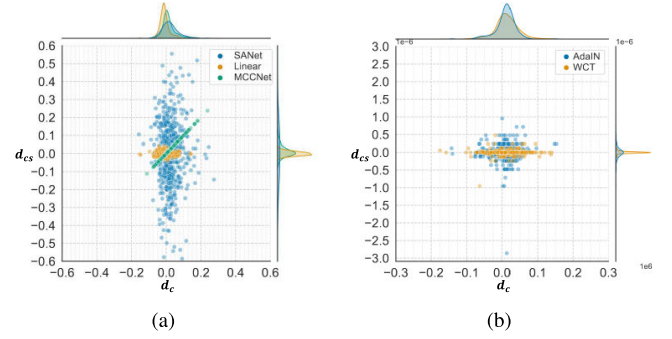


Fig. 3. Relationship for the difference in f_c and f_{cs} . The horizontal axis represents d_c , while the vertical axis represents d_{cs} . (a) Distribution of SANet, Linear, and MCCNet. (b) Distribution of AdaIN and WCT with extremely small values (10^{-6}).

as $f_{n,c}$, $f_{n+1,c}$ and the outputs of $\mathbb{T}(\cdot)$ as $f_{n,cs}$, $f_{n+1,cs}$. For the mentioned methods, we calculate the difference in $f_{n,c}$ and $f_{n+1,c}$ as d_c in each channel and observe the output difference d_{cs} of $f_{n,cs}$ and $f_{n+1,cs}$. Fig. 3 shows the results. WCT and SANet could take multilayer inputs for feature fusion; here, we consider the single-layer input only. Overall, d_{cs} of AdaIN and WCT is much smaller than that of d_c , while SANet and Linear could generate relatively consistent changes. However, the distribution of d_{cs} for SANet is different from d_c . Among these methods, only Linear tries to keep the distribution of d_{cs} and d_c . As discussed in Section V, the decoding results of Linear are the most stable, SANet ranks second, while AdaIN and WCT are the worst. Such observation motivates us to propose an antiflicker feature migration $\mathbb{T}(\cdot)$ that could keep a unanimous relationship for d_{cs} and d_c in each channel.

The prototype of antiflicker feature migration by which the output features could satisfy a strict linear alignment with content features is defined as follows:

$$f_{cs} = \mathbb{T}(f_c, f_s) = Kf_c \quad (6)$$

where K is an elementwise multiplication factor. The output f_{cs} achieves a linear relationship with f_c and can maintain the consistency of inputs naturally. Thus, we could measure the difference in $f_{n,cs}$ and $f_{m,cs}$ as

$$\begin{aligned} \|f_{n,cs} - Wf_{m,cs}\| &= \|Kf_{n,c} - WKf_{m,c}\| \\ &= |K| \cdot \|f_{n,c} - Wf_{m,c}\| < |K| \cdot \delta \end{aligned} \quad (7)$$

which means the feature-aligning migration can maintain the continuity among the input frames against flickering.

The prototype of feature-aligned migration focuses only on the linear alignment in the hidden representation space. Proposing a functional feature migration is a challenging task. AdaIN was involved in AST; thus, researchers have been working on an effective cross-domain feature fusion module to generate vivid stylization results. However, the transfer module $\mathbb{T}(\cdot)$ for most AST approaches is ill-conditioned, not satisfying (6) and (7). In this article, we propose a novel multichannel correlation as the feature migration module for AST tasks. As shown in Fig. 3(f), the proposed multichannel correlation could maintain a direct proportional relationship

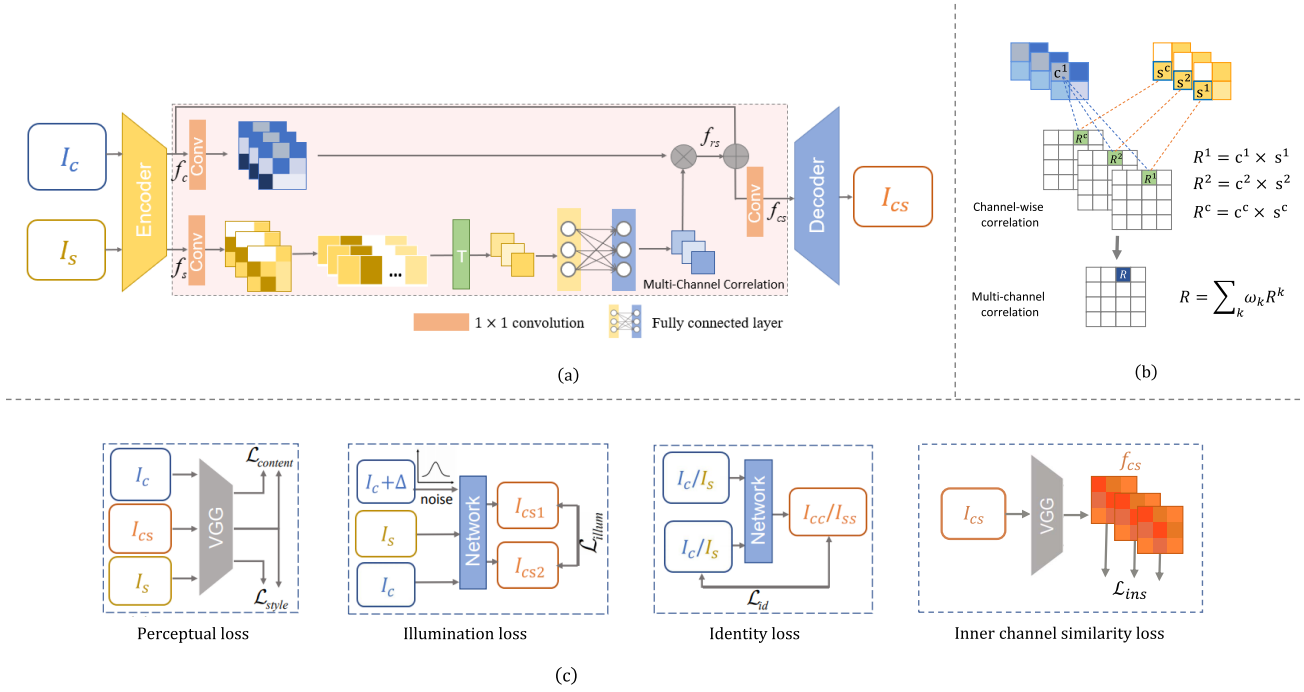


Fig. 4. Overall structure of MCCNet. (a) The green block represents the operation of the feature vector multiplied by the transpose of the vector. $\otimes$ and $\oplus$ represent the matrix multiplication and addition, respectively. I_c , I_s , and I_{cs} refer to the content, style, and generated images, respectively. (b) R is the element of CO. (a) Overall structure of MCCNet. (b) Correlation matrix CO. (c) Loss function.

between d_c and d_{cs} , which means the differences in the adjacent input frames could be well contained in the outputs from $\mathbb{T}(\cdot)$. A comparison between the relationship of d_c and d_{cs} for AdaIN and MCCNet shows that although both the approaches focus on generating a linear relationship for elements in a single content channel, the behavior among different channels is different.

IV. METHODOLOGY

In this section, we realize the prototype introduced in Section III-C. We propose the MCCNet adopting encoder-transform-decoder architecture while adhering to strict feature alignment in the transfer module, as shown in Fig. 2(f). The transfer module $\mathbb{T}(\cdot)$ of MCCNet uses the multichannel correlation through which f_{cs} satisfies the linear relationship with f_c as defined in (6). We start from the formulation and analysis of the multichannel correlation and feature alignment in Sections IV-A1 and then introduce the configuration of MCCNet in Section IV-B. The overall structure of the proposed MCCNet is shown in Fig. 4.

A. Feature Alignment via Multichannel Correlation

1) *Multichannel Correlation*: Cross-domain feature correlation has recently been studied for AST tasks [17], [35] as the transfer module. It fuses the content features f_c and style features f_s with an attention mask calculating between the two features. Benefiting from the strong expression ability of the attention mechanism, these methods could generate good visual results. However, when the content features f_c change slightly, a large-scale change can happen on output features.

As a result, it is not suitable for video style transfer because the input frames are slightly different from their neighbor frames all the time. Thus, the coherence of input content features cannot be migrated to generated features, and the stylized videos will present flickering effects.

In this work, we draw support from channelwise expressions and propose the multichannel correlation by exploiting the interchannel relationship of features for frame-based video stylization while maintaining the linear relationship between the content features and output features as (6).

For channel i , the content and style features are $f_c^i \in \mathbb{R}^{H \times W}$ and $f_s^i \in \mathbb{R}^{H \times W}$, respectively. We reshape them to $f_c^i \in \mathbb{R}^{1 \times N}$, $f_c^i = [c_1, c_2, \dots, c_N]$ and $f_s^i \in \mathbb{R}^{1 \times N}$, $f_s^i = [s_1, s_2, \dots, s_N]$, where $N = H \times W$. The channelwise correlation matrix between content and style is calculated by

$$\text{CO}^i = f_c^{iT} \otimes f_s^i. \quad (8)$$

The transposition of COrelation matrix (CO^{iT}) is shown in Fig. 4(b). Then, we rearrange the style features by

$$f_{rs}^i = f_s^i \otimes \text{CO}^{iT} = \|f_s^i\|_2 f_c^i \quad (9)$$

where $\|f_s^i\|_2 = \sum_{j=1}^N s_j^2$. Finally, the channelwise generated features are

$$f_{cs}^i = f_c^i + f_{rs}^i = (1 + \|f_s^i\|_2) f_c^i. \quad (10)$$

Wu and He [44] pointed out that a correlation exists among channels. To further enhance the stylization effects, we consider the correlation among content features by extending (9) to multichannel. The multichannel correlation in style features is important to represent style patterns (e.g., texture and color). We calculate the correlation between each content channel and

every style channel. The i th channel for generated features can be rewritten as

$$f_{cs}^i = f_c^i + f_{rs}^i = \left(1 + \sum_{k=1}^C w_k \|f_s^k\|_2\right) f_c^i \quad (11)$$

where C is the number of channels, and w_k represents the weights of the k th style channel.

We prove the importance of multichannel correlation in Section V by removing the weight parameter calculated via the multichannel correlation module, which means that the transformation applied to each channel of content features is completely the same. The stylization results show that not taking the multichannel correlation in style features into consideration will result in loss of abundant style patterns and result in style collapse.

2) *Feature Alignment*: (11) could be naturally deduced to (6) by setting $K = (1 + \sum_{k=1}^C w_k \|f_s^k\|_2)$. When the style reference is fixed, the value of K is a constant that represents the style information learned by the encoder. We can conclude that adopting multichannel correlation as transfer module $\mathbb{T}(\cdot)$ can help generate stylized features that are strictly aligned with content features.

Therefore, the coherence of input videos is naturally maintained to output videos, and slight changes (e.g., object motions) in adjacent frames cannot lead to violent changes in the stylized video. Moreover, through the correlation calculation between content and style representations, the style patterns are adjusted according to content distribution. The adjustment helps generate stylized results with stable content structures and appropriate style patterns.

Unlike AdaIN, multichannel correlation satisfies a strict linear relationship between f_c^i and f_{cs}^i without bias terms and takes local features into consideration. Unlike WCT and Linear, multichannel correlation uses inner channel information because it calculates attention on all the pixels within a single feature map on the basis of multichannel correlation. Unlike SANet, the calculation of cross-channel correlation makes the transfer module of MCCNet well-conditioned. The usage of cross-channel information also improves the quality of stylization.

B. Network Structure and Training

In this section, we introduce how to involve multichannel correlation in the encoder–transform–decoder-based stylization framework [Fig. 4(a)]. Given f_c and f_s , we feed them to the 1×1 convolution layer. We then stretch $f_s \in \mathbb{R}^{C \times H \times W}$ to $f_s \in \mathbb{R}^{C \times 1 \times N}$, and $f_s f_s^T = [\|f_s^1\|_2, \|f_s^2\|_2, \dots, \|f_s^C\|_2]$ is obtained through covariance matrix calculation. Next, we add a fully connected layer to weigh the different channels in $f_s f_s^T$. Through matrix multiplication, we weigh each content channel with a compound of different channels in $f_s f_s^T$ to obtain f_{rs} . Then, we add f_c and f_{rs} to obtain f_{cs} as defined in (11). Finally, we feed f_{cs} to a 1×1 convolutional layer and decode it to obtain the generated image I_{cs} .

As shown in Fig. 4(c), our network is trained by minimizing the loss function defined as

$$\begin{aligned} \mathcal{L} = & \lambda_{\text{content}} \mathcal{L}_{\text{content}} + \lambda_{\text{style}} \mathcal{L}_{\text{style}} \\ & + \lambda_{\text{id}} \mathcal{L}_{\text{id}} + \lambda_{\text{illum}} \mathcal{L}_{\text{illum}} \\ & + \lambda_{\text{ins}} \mathcal{L}_{\text{ins}}. \end{aligned} \quad (12)$$

The total loss function includes perceptual losses $\mathcal{L}_{\text{content}}$ and $\mathcal{L}_{\text{style}}$, identity loss $\mathcal{L}_{\text{id}}$, illumination loss $\mathcal{L}_{\text{illum}}$, and inner channel similarity loss $\mathcal{L}_{\text{ins}}$ in the training procedure. The weights λ_{content} , λ_{style} , λ_{id} , λ_{illum} , and λ_{ins} are set to 4, 15, 70, 3000, and 3, respectively, to eliminate the impact of magnitude differences.

1) *Perceptual Loss*: We use a pretrained very deep network (VGG)19 [45] to extract content and style features and compute the content and style perceptual loss similar to AdaIN [15]. In our model, we use layer conv4_1 to calculate the content perceptual loss, and we use layers conv1_1, conv2_1, conv3_1, and conv4_1 to calculate the style perceptual loss. The content perceptual loss $\mathcal{L}_{\text{content}}$ is used to minimize the content differences between generated images and content images, where

$$\mathcal{L}_{\text{content}} = \|\phi_4(I_{cs}) - \phi_4(I_c)\|_2. \quad (13)$$

The style perceptual loss $\mathcal{L}_{\text{style}}$ is used to minimize the style differences between generated images and style images

$$\mathcal{L}_{\text{style}} = \sum_{i=1}^L \mathcal{L}_{\text{style}}^i \quad (14)$$

$$\begin{aligned} \mathcal{L}_{\text{style}}^i = & \|\mu(\phi_i(I_{cs})) - \mu(\phi_i(I_s))\|_2 \\ & + \|\sigma(\phi_i(I_{cs})) - \sigma(\phi_i(I_s))\|_2 \end{aligned} \quad (15)$$

where $\phi_i(\cdot)$ denotes the features extracted from the i th layer in a pretrained VGG19, $\mu(\cdot)$ denotes the mean of features, and $\sigma(\cdot)$ denotes the variance of features.

2) *Inner Channel Similarity Loss*: The proposed MCCNet pays attention to fusing the multichannel information but neglects the correlation inner each channel. Moreover, the strict alignment between f_{cs} and f_c is achieved at the cost of abandoning nonlinear operations such as softmax, which could normalize the distribution of data within a channel, and results in an unsatisfactory style pattern. An inner similarity is raised because it does not hurt the alignment while strengthening the consistency of generated features within each channel, thus avoiding the appearance of disharmonious areas. The loss can be formulated as follows:

$$\mathcal{L}_{\text{ins}} = \sum_{c=1}^C \text{Inner}_{c,i} \quad (16)$$

where C is the channels of feature, and the inner channel similarity $\text{Inner}_{c,i}$ is defined as

$$\text{Inner}_{c,i} = \arg \min_i \sum_{j=1}^{h*w} \left(1 - \frac{f_i \cdot f_j}{\|f\|_F^2}\right) \quad (17)$$

where f represents the generated features, and $h * w$ is their resolution.

TABLE I

STATISTICS OF INFERENCE SPEED AND QUANTITATIVE COMPARISON WITH SOTA METHODS. THE RESULTS OF USER STUDY REPRESENT THE AVERAGE PERCENTAGE OF CASES IN WHICH THE RESULT OF THE CORRESPONDING METHOD IS PREFERRED COMPARED WITH OURS

Method		Ours	Stytr ²	AdaAttN	ArtFlow	CompoundVST	MAST	Linear	SANet	AAMS	WCT	AdaIN	NST
Inference time (seconds/image)	256	0.013	0.040	0.054	0.255	0.049	0.030	0.010	0.015	2.074	0.451	0.007	19.528
	512	0.015	0.103	0.071	0.664	0.098	0.096	0.013	0.019	2.173	0.579	0.008	37.211
	1024	0.019	0.605	0.084	2.332	0.285	0.506	0.022	0.021	2.456	1.008	0.009	106.372
Classification accuracy (%)↑	Content	36.5	24.8	28.2	37.5	–	27.8	49.2	28.1	38.4	22.0	40.8	32.0
	Style	63.0	62.5	53.5	42.1	–	57.5	50.7	67.4	57.8	61.5	58.7	57.7
User study (% of wins)	Image	–	37.7	31.4	27.9	–	22.3	33.2	26.0	22.4	22.8	30.0	28.0
	Video	–	23.9	28.5	25.9	29.6	19.8	27.3	18.4	15.8	18.7	18.7	24.4

3) *Identity Loss*: We adopt identity loss to constrain the mapping relationship between style features and content features and help our model maintain the content structure without losing the richness of the style patterns. The identity loss $\mathcal{L}_{id}$ is defined as

$$\mathcal{L}_{id} = \|I_{cc} - I_c\|_2 + \|I_{ss} - I_s\|_2 \quad (18)$$

where I_{cc} denotes the generated results using a common natural image as content image and style image, and I_{ss} denotes the generated results using a common painting as content image and style image.

4) *Illumination Loss*: For a video sequence, the illumination may change slightly in a way that is difficult for humans to discover by humans. The illumination variation in video frames could influence the final transfer results and result in flickering. The output video suffers from severe light flicker even when the illumination does not change in the input video.

Therefore, we add random Gaussian noise to the content images to simulate light and propose the illumination loss as

$$\mathcal{L}_{illum} = \|G(I_c, I_s) - G(I_c + \Delta, I_s)\|_2 \quad (19)$$

where $G(\cdot)$ is our generation function, and $\Delta \sim \mathcal{N}(0, \sigma^2 I)$. With illumination loss, our method can be robust to complex light conditions in the input video.

V. EXPERIMENTS

A desired video stylization method could not only promote the stability of stylized video but also generate fascinating single frame style transfer results. Typical video stylization methods use temporal constraint and optical flow to avoid flickering in the generated videos. Our method focuses on promoting the stability of the transform operation in the AST model on the basis of a single frame. Thus, we test the proposed MCCNet on image and video stylization tasks. The following state-of-the-art (SOTA) AST methods are selected for comparison: NST-2016 [5], AdaIN-2017 [15], WCT-2017 [16], Linear-2019 [26], SANet-2019 [17], AAMS-2019 [34], MAST-2020 [35], CompoundVST-2020 [38], ArtFlow-2021 [46], AdaAttN-2021 [47], and Stytr²-2022 [48]. NST [5] is the representative work for the optimization-based AST methods. AdaIN [15] and WCT [16] are the two most classic encoder-transform-decoder-based AST approaches. The rest are open-source SOTAs for image/video AST tasks. For SANet and Stytr², we resize the content frames to fit its limitation of input size, which may reduce the visual effects.

In the rest of this section, we start from the training details of the proposed approach, and then evaluate the image (frame) stylization effects and analyze the rendered videos. The comparisons with our conference version [28] will be shown in Section V-D.

A. Implementation Details and Statistics

We use Microsoft Common Objects in COntext (MS-COCO) [49] and WikiArt [50] as the content and style image datasets for network training. No video clips are involved for training. In the training stage, the images are randomly cropped to 256×256 pixels. At the inference stage, images with an arbitrary size are acceptable. The encoder is a pretrained VGG19 network. The decoder is a mirror version of the encoder, except for the parameters that need to be trained. The training batch size is 8, and the whole model is trained through 160 000 steps.

1) *Timing Information*: We measure our inference time for the generation of an output image and compare the result with those of SOTA methods using 16 G TitanX graphics processing unit (GPU). The optimization-based method NST [5] is trained for 30 epochs. Table I shows the inference times of different methods using three scales of image size. The proposed MCCNet is much faster than NST, WCT, AAMS, CompoundVST, MAST, and ArtFlow and can achieve a real-time transfer speed that is comparable to that of AdaIN, Linear, and SANet for efficient video style transfer. Compared with our conference version [28], the improvement is brought with no inference burden and does not consume additional time.

B. Image Style Transfer

1) *Qualitative Analysis*: CompoundVST is not selected for image stylization comparison because it is designed for video style transfer. When applied to a single image without global regulation, the outputs of CompoundVST are not readable. The comparisons of image stylizations are shown in Fig. 5. Although the proposed multichannel correlation focuses on generating stable results, MCCNet does not cause a decline in the stylization results. On the basis of the optimized training mechanism, NST may introduce failure results (14th row) and cannot easily achieve a trade-off between the content structure and style patterns in the rendered images. In addition to crack effects caused by oversimplified calculation in AdaIN, the absence of correlation between different channels causes a poor transfer of style color distribution (13th row). As for WCT, the local content structures of the generated results

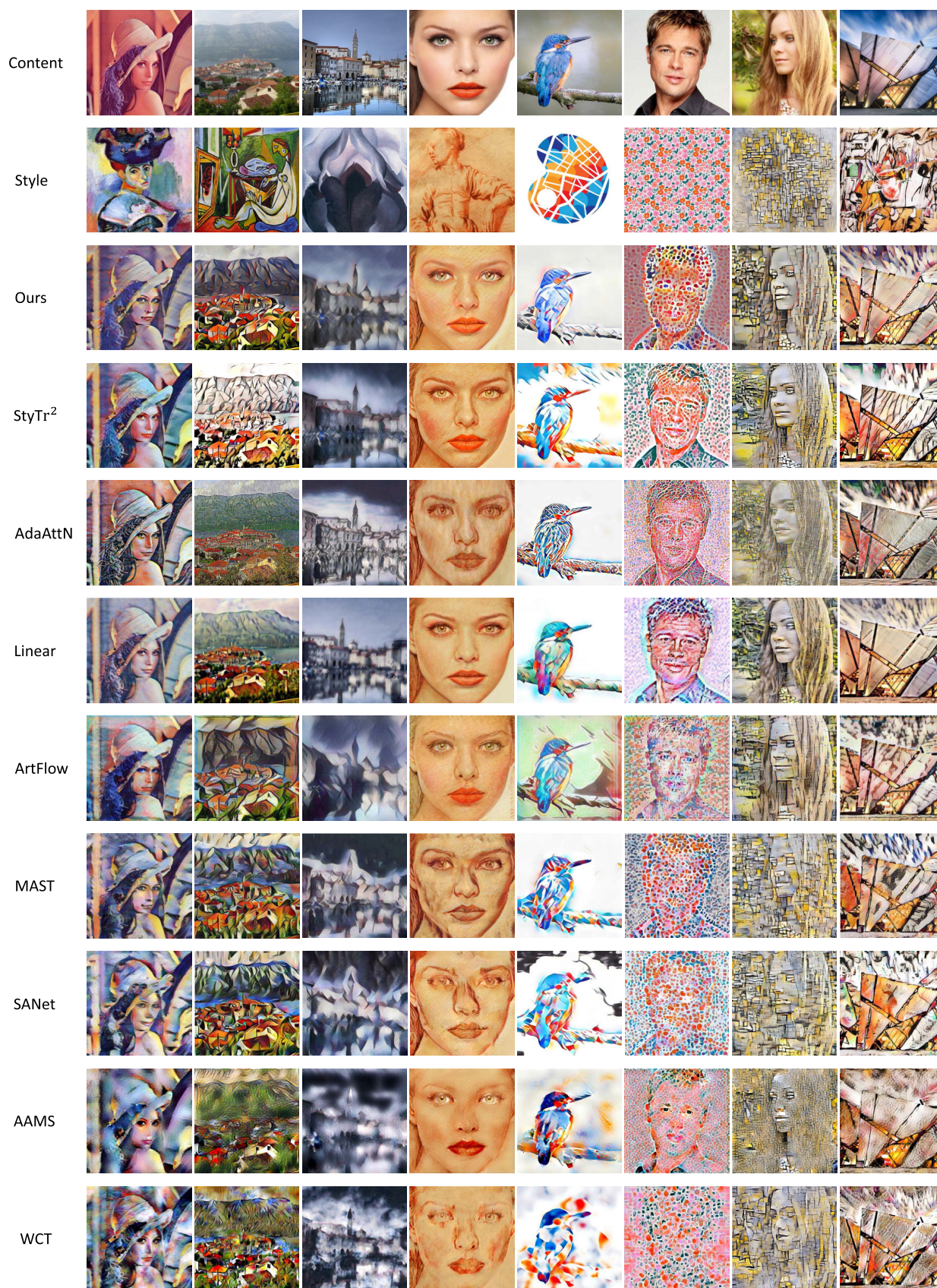


Fig. 5. Comparisons of image style transfer results (recommend printing in color or viewing online). The first row shows the content images, and the second row shows the style images. The remaining columns are stylized results of different methods.

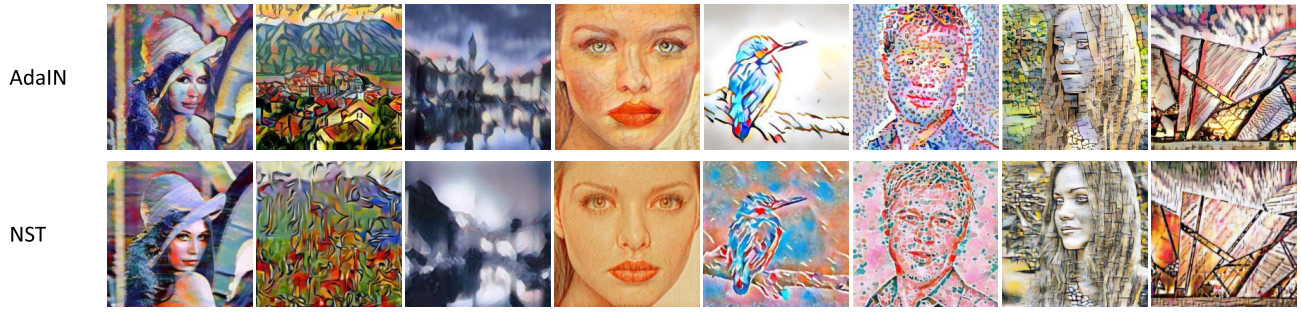


Fig. 5. (Continued.) Comparisons of image style transfer results (recommend printing in color or viewing online). The first row shows the content images, and the second row shows the style images. The remaining columns are stylized results of different methods.

are damaged due to the global parameter transfer method (12th row). Although the attention map in AAMS helps make the main structure exact, the patch splicing trace affects the overall generated effect (11th row). Some style image patches are transferred into the content image directly by SANet and damage the content structures (10th row). By learning a transformation matrix for style transfer, the process of Linear is too simple to acquire adequate style textural patterns for rendering (6th row). MAST may introduce unexpected style patterns in rendered results (8th row). Artflow keeps the content well but generates some unharmonious style patterns (7th row). Some stylized results of Stytr² are unharmonious since the Vit backbone does not have a local receptive field for style patterns and has a bias to shape (4th row). Adding extra stability constraints, AdaAttN pays too much attention to content edges (5th row).

Our network calculates the multichannel correlation of content and style features and rearranges style features on the basis of the correlation. The rearranged style features fit the original content features, and the fused generated features consist of clear content structures and controllable style pattern information. Therefore, our method can achieve satisfactory stylized results.

2) *Quantitative Analysis*: We train a content and a style classification model to assess the performance of different style transfer networks by considering their ability to maintain content structure and style migration, respectively. We generate several stylized images using the different aforementioned style transfer methods. Then, we feed the stylized images generated by different methods into the style and content classification models. High-accuracy style classification indicates that the style transfer network can easily learn to obtain effective style information while high-accuracy content classification indicates that the style transfer network can easily learn to maintain the original content information. From Table I, we can conclude that AAMS, AdaIN, and MCCNet can achieve a good trade-off between content and style. However, our network is superior to the two methods in terms of visual effects.

3) *User Study*: We conducted user studies to compare the stylization effect of our method with those of the aforementioned SOTA methods. We selected 20 content images and 20 style images to generate 400 stylized images using

different methods. First, we showed participants a pair of content and style images. Second, we showed them two stylized results generated by our method and a random contrast method. Finally, we asked the participants which stylized result has the best rendered effects by considering the integrity of the content structures and the visibility of the style patterns. We collected 3273 votes from 72 participants and presented the voting results in Table I. Overall, our method can achieve the best image stylization effect. Linear, AdaAttN, and Stytr² are comparable to MCCNet in terms of visual quality.

C. Video Style Transfer

1) *Video Stylization*: We compare the video stylization results of our proposed method with the SOTA video style transfer methods, including CompoundVST [38], Linear [26], and AdaAttN [47]. MCCNet draws support from multichannel correlation calculation and generates features with clear content structures and controllable style pattern information, achieving stable video stylization without damaging the stylization effect, as shown in Fig. 6. However, all the three compared methods introduce extra stability constraints to avoid the flickering effects, which deteriorates the stylization results. Furthermore, depending on sequence-level global calculation, CompoundVST could not be applied to long video clips. The maximum video duration that CompoundVST can handle for 1024×1024 inputs is 2.88 s in our testing environment, while MCCNet does not have such a limitation. Although Linear and AdaAttN can also generate a relatively coherent video, the resultant continuity is influenced by nonlinear operation in deep CNNs. More visual comparisons of the consistency of the generated videos are given in the supplementary materials.

2) *Video Consistency*: We quantitatively analyze the temporal consistency of the generated videos using optical flow as [38]. Given two adjacent frames I_c^t and I_c^{t-1} in a T-frame input clip and I_{cs}^t and I_{cs}^{t-1} in a T-frame rendered clip, the temporal loss is defined as

$$L_{\text{temporal}} = \text{average} \left(\left\| O \circ \left(W_{I_c^{t-1} \rightarrow I_c^t} (I_{cs}^{t-1}) - I_{cs}^t \right) \right\| \right) \quad (20)$$

where O is an occlusion mask

$$O = \left| W_{I_c^{t-1} \rightarrow I_c^t} (I_{cs}^{t-1}) - I_{cs}^t \right| > 10. \quad (21)$$

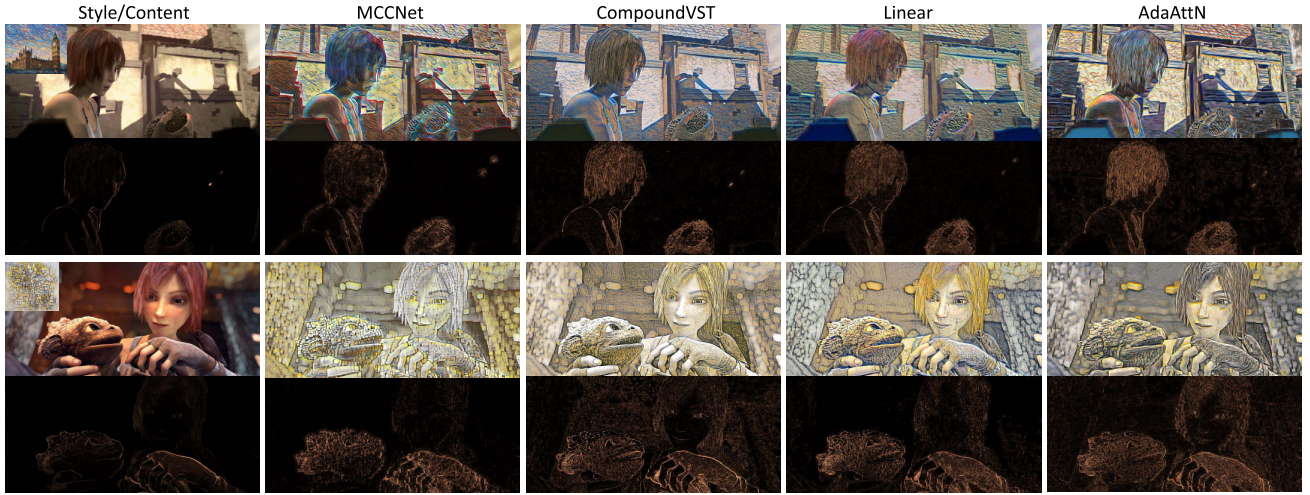


Fig. 6. Comparisons of video style transfer results. The first column is the input video frames and style images, and the rest of the columns are the stylization results of different methods. The first row of each group shows the stylization results, and the second row shows the heat maps which visualize the difference in two stylized adjacent frames.

TABLE II
QUANTITATIVE EVALUATION OF TEMPORAL CONSISTENCY ON 14 RENDERED CLIPS

	Ours	Stytr ²	AdaAttN	Linear	CompoundVST	NST	ArtFlow	SANet	MAST	WCT	AdaIN	AAMS
$\mathcal{L}_{temporal} \downarrow$	0.0120	0.0131	0.0190	0.0128	0.0132	0.0157	0.0168	0.0183	0.0200	0.0204	0.0203	0.0209

The mask eliminates the negative effects brought by inaccurate optical flow estimation. From the statistic in Table II, we can conclude that our method can yield the best video results with high consistency. Linear and CompoundVST can also generate relatively coherent videos. However, the visual effects of Linear and AdaAttN are not good, and the proposed MCCNet is about $4\times$ to $10\times$ faster than CompoundVST.

3) *User Study*: We conducted user studies to compare the video stylization effects of our method with those of the aforementioned SOTAs. First, we showed the participants an input video clip and a style image. Second, we showed them two stylized video clips generated by our method and a random contrast method. Finally, we asked the participants which stylized video clip was the most satisfactory by considering the stylized effect and the stability of the videos. Each user answered 10–14 questions. We collected 1000 votes from 78 participants and presented the voting results in Table I. Overall, our method can achieve the best video stylization effect.

D. Ablation Study

We conduct ablation studies to investigate the contribution of each term to the quality of style transfer. We research the impact brought by multichannel correlation, illumination loss, and inner channel similarity loss. Table III shows the overall results for the influence of these modules on video stability. Multichannel correlation and inner channel similarity have a positive effect on transferring style patterns, and illumination plays an important role in eliminating the flickering caused by the changing illumination condition.

1) *Inner Channel Similarity Loss*: MCCNet pays attention to interchannel correlation and achieves good stylization

TABLE III
ABLATION STUDY ON VIDEO STABILITY

MCC module	$\mathcal{L}_{illum}$	$\mathcal{L}_{ims}$	$\mathcal{L}_{temporal} \downarrow$
-	✓	✓	0.0125
✓	-	✓	0.0131
✓	✓	-	0.0127
✓	✓	✓	0.0120

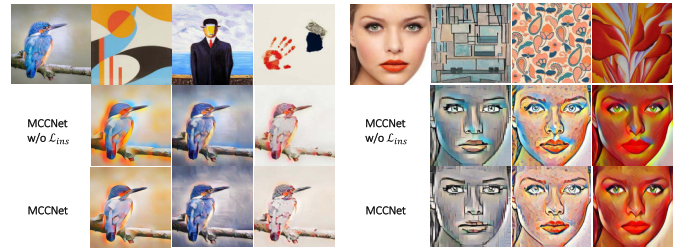


Fig. 7. Comparisons of MCCNet with and without $\mathcal{L}_{ims}$. Without $\mathcal{L}_{ims}$, the results are the same as those in [28].

results. However, it achieves strict alignment between f_{cs} and f_s at the cost of abandoning the softmax operation, which would cause a severe highlight effect around the object. To avoid hurting the alignment, we do not add any additional operation to the network but add an inner channel similarity loss as a supplementary to inner channel correlation. Without $\mathcal{L}_{ims}$, MCCNet is the same as our conference version. Overall, the results with and w/o $\mathcal{L}_{ims}$ may be similar. However, with the proposed $\mathcal{L}_{ims}$, the highlight effect around the objects is eliminated. The visual comparisons in Fig. 7 show the importance of inner channel correlation.



Fig. 8. Ablation study of removing the illumination loss. The bottom row shows heat maps used to visualize the differences between two input frames.

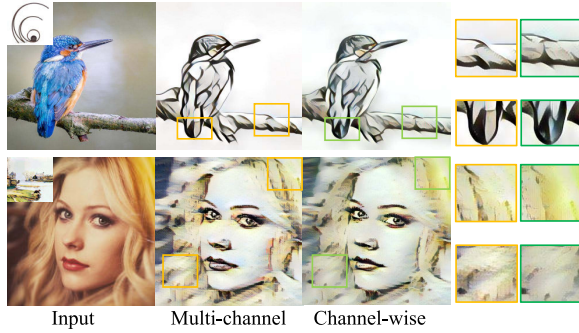


Fig. 9. Ablation study of channel correlation. Without multichannel correlation, the stylized results present fewer style patterns (e.g., hair of the woman) and may maintain the original color distribution (the blue color in the bird's tail).

Moreover, the removal of the highlight effect brought by $\mathcal{L}_{ins}$ is favorable for video stability because it eliminates local abnormal values that may differ from two adjacent stylized frames. As shown in Table III, without $\mathcal{L}_{ins}$, $\mathcal{L}_{temporal}$ increased from 0.0120 to 0.0127.

2) *Illumination Loss*: The illumination loss is proposed to eliminate the impact of video illumination variation. We remove the illumination loss in the training stage and compare the results with ours in Fig. 8. Without illumination loss, the differences between the two stylized frames increase, with the mean $\mathcal{L}_{temporal}$ being 0.0131 as shown in Table I. With illumination loss, our method can be flexibly applied to videos with complex light conditions.

3) *Channel Correlation*: Our network is based on multi-channel correlation calculation, which is shown in Fig. 4(b). To analyze the influence of multichannel correlation on stylization effects, we change the model to calculate single-channel correlation without considering the relationship between style channels. Fig. 9 shows the results. Through single channel calculation, the stylized results show few style patterns (e.g., hair of the woman) and may maintain the original color distribution (the blue color in the bird's tail). Meanwhile, the style patterns are effectively transferred by considering the multichannel information in the style features.

Moreover, we analyze the effect of multichannel correlation for video stylization task in Fig. 10 and Table III. With channelwise correlation, the coherence between the two stylized frames is reduced, with the mean $\mathcal{L}_{temporal}$ being 0.0125. With multichannel correlation, the generated features have a sense of the global information instead of the only corresponding

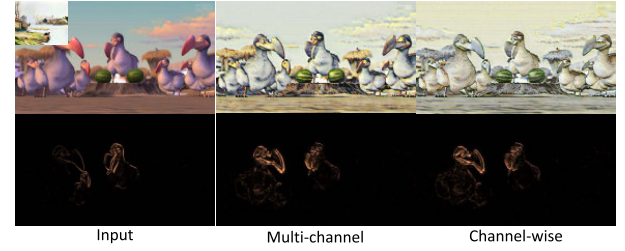


Fig. 10. Ablation study of channel correlation on video stylization. Without multichannel correlation, the stylized results present fewer style patterns and increased difference between two frames.

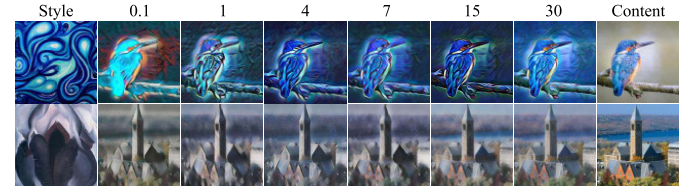


Fig. 11. Training network with different $\lambda_{content}$.

channel, which not only helps grasp more style patterns but is also favorable for video stability. The results have more exquisite style patterns while achieving better stability.

4) *Hyperparameter Analysis*: The hyperparameters λ_{id} , λ_{illum} , and $\mathcal{L}_{ins}$ in (12) are set to eliminate the impact of magnitude differences. For example, the illumination loss $\mathcal{L}_{illum}$ is small, thus needing a large weight for fast training. Different settings of these parameters will not influence the final visual results but lead to different training times. The content loss weight $\lambda_{content}$ and the style loss weight λ_{style} are responsible for the stylization level of the generated images. We train our models with fixed λ_{style} and different $\lambda_{content}$. As shown in Fig. 11, when $\lambda_{content}$ is small, the stylization results are intense and lose content information. In contrast, when $\lambda_{content}$ is equal or larger than λ_{style} , the style patterns are not transferred sufficiently.

E. Discussion

1) *Feature-Aligned Migration*: As mentioned in Section III-C, a strict linear alignment between f_{cs} and f_s can maintain the coherency of input frames naturally. The feature migration $\mathbb{T}(\cdot)$ of AdaIN cannot strictly satisfy the requirement and an additional bias term exists. We further verify (6) by dropping the bias term in (3) and turning AdaIN into a channelwise strictly aligning method. The modified style transfer module is defined as follows:

$$f_{cs}^i = \sigma(f_s^i) \frac{f_c^i - \mu(f_c^i)}{\sigma(f_c^i)}. \quad (22)$$

By replacing $\mathbb{T}(\cdot)$ in AdaIN as (22), we train AdaIN-variation following the same configuration as AdaIN. The temporal consistency of AdaIN-variation is 0.0175. Unlike vanilla AdaIN (0.0203), the feature-aligned migration in (22) improves the temporal consistency of the generated results. The residual maps in Fig. 12 also show that AdaIN-variation successfully transfers the coherency of the input video to the output videos. However, directly dropping the bias item in AdaIN will cause

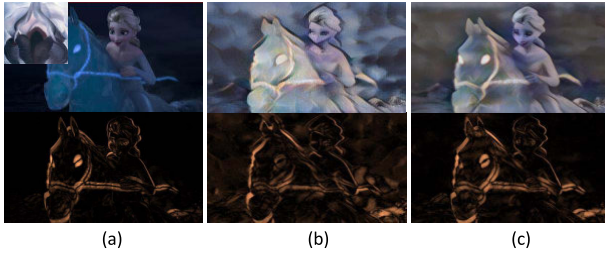


Fig. 12. Residual image of AdaIN and its variation using frames from *Frozen*. (a) Input. (b) AdaIN. (c) AdaIN-variation.



Fig. 13. Image stylization results of one content image and multiple similar style images.

f_{cs} to lose rich style information. As shown in Fig. 12, the results of AdaIN-variation are less stylized compared with those of AdaIN. Such results show that the design of feature migration $\mathbb{T}(\cdot)$ is a challenging task. Both the feature alignment and the stylization effect should be taken into consideration.

We analyze the influence of the feature-aligned migration module on content feature overfitting. The prototype of antiflicker feature migration focuses on the stability of the generated videos only. Without a proper design of the migration module, the stylized network may overfit the content features. As shown in Fig. 13, given the same content input, the stylized effects of AdaIN-variation are not visually appealing due to strict feature migration. In contrast, the MCCNet could generate various visual results with the guide of different style images.

2) *Network Depth*: Linear [26] argues that the nonlinear layers in the decoder will introduce temporal inconsistency. When their method is applied to video style transfer, they should reduce the depth of decoder layers to acquire stable transferred videos. However, the shallow AE cannot learn enough style information, which leads to unsatisfactory results. We use a shallow AE up to *relu3-1* instead of *relu4-1* to discuss the effects of the convolution operation of the decoder on our model. As shown in Fig. 14, for image style transfer, the shallower network cannot generate results with vivid style patterns. For video style transfer, we use heat maps to visualize the differences between two video frames. As shown in Fig. 15, the depth of the network exerts little impact on the

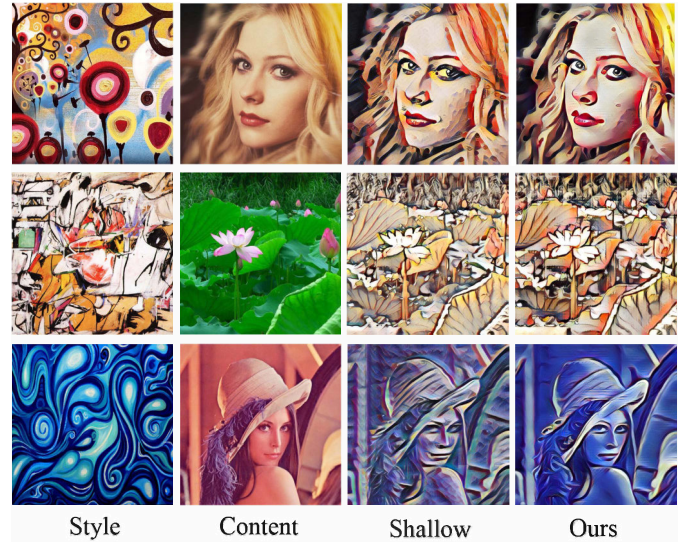


Fig. 14. Ablation study of network depth. A shallower network will reduce the visual effects of the generated results.

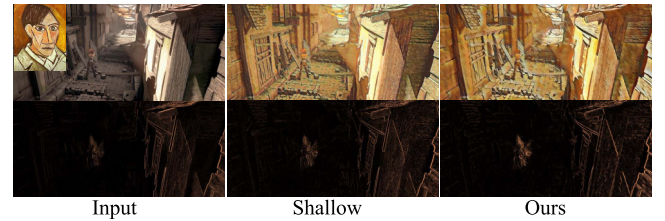


Fig. 15. Ablation study of network depth. The heat maps in the second row show the differences between two video frames in the first row.

coherence of the stylized video. This phenomenon suggests that the coherence of stylized frame features can be well-transferred to the generated video despite the convolution operation of the decoder.

VI. CONCLUSION

In this work, we analyze the cause of flickering effects for the encode–transform–decoder-based AST approaches and propose MCCNet for stable arbitrary video style transfer. The proposed network can migrate the coherence of input videos to stylized videos, thereby guaranteeing rendered videos' stability. Meanwhile, MCCNet with inner channel similarity loss can generate stylized results with vivid style patterns and detailed content structures by analyzing the multichannel correlation between content and style features. Moreover, the illumination loss improves the stability of the generated videos under complex light conditions. Extensive experiments demonstrate that the proposed method achieves pleasing stylized images and videos at a high inference speed.

REFERENCES

- [1] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4990–4998.
- [2] M. Zhang, N. Wang, Y. Li, and X. Gao, "Neural probabilistic graphical model for face sketch synthesis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2623–2637, Jul. 2020.
- [3] S. Jiang, J. Li, and Y. Fu, "Deep learning for fashion style generation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 9, pp. 4538–4550, Sep. 2022.

- [4] G. Kwon and J. C. Ye, "CLIPstyle: Image style transfer with a single text condition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18062–18071.
- [5] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.
- [6] J. Justin, A. Alexandre, and F.-F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Amsterdam, The Netherlands: Springer, 2016, pp. 694–711.
- [7] C. Peng, X. Gao, N. Wang, D. Tao, X. Li, and J. Li, "Multiple representations-based face sketch-photo synthesis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2201–2215, Nov. 2016.
- [8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (CVPR)*, Oct. 2017, pp. 2223–2232.
- [9] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Munich, Germany: Springer, 2018, pp. 172–189.
- [10] Y. Jing et al., "Dynamic instance normalization for arbitrary style transfer," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 4369–4376.
- [11] H. Mun, G.-J. Yoon, J. Song, and S. M. Yoon, "Texture preserving photo style transfer network," *IEEE Trans. Multimedia*, vol. 24, pp. 3823–3834, 2022.
- [12] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, "Transductive face sketch-photo synthesis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 9, pp. 1364–1376, May 2013.
- [13] Z. Ma et al., "Dual-affinity style embedding network for semantic-aligned image style transfer," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 2, 2022, doi: [10.1109/TNNLS.2022.3143356](https://doi.org/10.1109/TNNLS.2022.3143356).
- [14] S. Ruan et al., "DAE-GAN: Dynamic aspect-aware GAN for text-to-image synthesis," in *Proc. IEEE Int. Conf. Comput. Vis. (CVPR)*, Oct. 2021, pp. 13960–13969.
- [15] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1501–1510.
- [16] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 386–396.
- [17] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5880–5888.
- [18] Y. Wang, J. Hsiao, O. Sorkine, and T. Lee, "Scalable and coherent video resizing with per-frame optimization," *ACM Trans. Graph. (TOG)*, vol. 30, no. 4, pp. 1–8, Jul. 2011.
- [19] T.-C. Wang et al., "Video-to-video synthesis," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 1152–1164.
- [20] F.-L. Zhang, C. Barnes, H.-T. Zhang, J. Zhao, and G. Salas, "Coherent video generation for multiple hand-held cameras with dynamic foreground," *Comput. Vis. Media*, vol. 6, no. 3, pp. 291–306, Sep. 2020.
- [21] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *Proc. German Conf. Pattern Recognit. (GCPR)*. Hannover, Germany: Springer, 2016, pp. 26–36.
- [22] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, "Coherent online video style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1105–1114.
- [23] S. Liu and T. Zhu, "Structure-guided arbitrary style transfer for artistic image and video," *IEEE Trans. Multimedia*, vol. 24, pp. 1299–1312, 2022.
- [24] W. Gao, Y. Li, Y. Yin, and M.-H. Yang, "Fast video multi-style transfer," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 3222–3230.
- [25] H. Huang et al., "Real-time neural style transfer for videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 783–791.
- [26] X. Li, S. Liu, J. Kautz, and M.-H. Yang, "Learning linear transformations for fast image and video style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3809–3817.
- [27] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [28] Y. Deng, F. Tang, W. Dong, H. Huang, C. Ma, and C. Xu, "Arbitrary video style transfer via multi-channel correlation," in *Proc. 35th AAAI Conf. Artif. Intell. (AAAI)*, 2021, pp. 1210–1217.
- [29] M. Lin, F. Tang, W. Dong, X. Li, C. Xu, and C. Ma, "Distribution aligned multimodal and multi-domain image stylization," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 3, pp. 1–17, Jul. 2021.
- [30] Y. Zhang et al., "Domain enhanced arbitrary image style transfer via contrastive learning," in *Proc. ACM SIGGRAPH Conf.*, 2022, pp. 1–8.
- [31] N. Huang, F. Tang, W. Dong, and C. Xu, "Draw your art dream: Diverse digital art synthesis with multimodal guided diffusion," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 1085–1094.
- [32] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–11.
- [33] Z. Wang et al., "Diversified arbitrary style transfer via deep feature perturbation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7789–7798.
- [34] Y. Yao, J. Ren, X. Xie, W. Liu, Y.-J. Liu, and J. Wang, "Attention-aware multi-stroke style transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1467–1475.
- [35] Y. Deng, F. Tang, W. Dong, W. Sun, F. Huang, and C. Xu, "Arbitrary style transfer via multi-adaptation network," in *Proc. 28th ACM Int. Conf. Multimedia*, New York, NY, USA, Oct. 2020, pp. 2719–2727.
- [36] H. Wei, Y. Deng, F. Tang, X. Pan, and W. Dong, "A comparative study of CNN- and transformer-based visual style transfer," *J. Comput. Sci. Technol.*, vol. 37, no. 3, pp. 601–614, 2022.
- [37] X. Chen, Y. Zhang, Y. Wang, H. Shu, C. Xu, and C. Xu, "Optical flow distillation: Towards efficient and stable video style transfer," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Glasgow, U.K.: Springer, 2020, pp. 614–630.
- [38] W. Wang, J. Xu, L. Zhang, Y. Wang, and J. Liu, "Consistent video style transfer via compound regularization," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 12233–12240.
- [39] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
- [40] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
- [41] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.
- [42] J. Fu, J. Liu, J. Jiang, Y. Li, Y. Bao, and H. Lu, "Scene segmentation with dual relation-aware attention network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2547–2560, Jun. 2021.
- [43] Y. Ji, H. Zhang, Z. Jie, L. Ma, and Q. M. J. Wu, "CASNet: A cross-attention Siamese network for video salient object detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2676–2690, Jul. 2021.
- [44] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Munich, Germany: Springer, 2018, pp. 3–19.
- [45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14.
- [46] J. An, S. Huang, Y. Song, D. Dou, W. Liu, and J. Luo, "ArtFlow: Unbiased image style transfer via reversible neural flows," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 862–871.
- [47] S. Liu et al., "AdaAttN: Revisit attention mechanism in arbitrary neural style transfer," in *Proc. IEEE Int. Conf. Comput. Vis. (CVPR)*, Oct. 2021, pp. 6649–6658.
- [48] Y. Deng et al., "StyTr²: Image style transfer with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11326–11336.
- [49] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Zürich, Switzerland: Springer, 2014, pp. 740–755.
- [50] F. Phillips and B. Mackintosh, "Wiki art gallery, Inc.: A case for critical thinking," *Issues Accounting Educ.*, vol. 26, no. 3, pp. 593–608, 2011.



Xiaoyu Kong received the B.Sc. degree in computer science from Jilin University, Changchun, China, in 2021. He is currently pursuing the master's degree from the Harbin Institute of Technology, Shenzhen, China.

His research interests include computer vision and machine learning.



Yingying Deng received the B.Sc. degree in automation from the University of Science and Technology, Beijing, China, in 2017. She is currently pursuing the Ph.D. degree with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing.

Her research interests include computational visual media and machine learning.



Yongyong Chen received the B.S. and M.S. degrees from the Shandong University of Science and Technology, Qingdao, China, in 2014 and 2017, respectively, and the Ph.D. degree from the University of Macau, Macau, China, in 2020.

He is currently an Assistant Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include image processing, data mining, and computer vision.



Fan Tang received the B.Sc. degree in computer science from North China Electric Power University, Beijing, China, in 2013, and the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, in 2019.

He is an Assistant Professor with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer graphics, computer vision, and machine learning.



Zhenyu He (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science, Hong Kong Baptist University, Hong Kong, in 2007.

From 2007 to 2009, he worked as a Post-Doctoral Researcher with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. He is currently a Full Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include machine learning, computer vision, image processing, and pattern recognition.



Weiming Dong (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science from Tsinghua University, Beijing, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science from the University of Lorraine, Nancy, France, in 2007.

He is a Professor with the Sino-European Lab in Computer Science, Automation and Applied Mathematics (LIAMA) and National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing. His research

interests include image synthesis and image recognition.

Dr. Dong is a member of the Association for Computing Machinery (ACM).



Changsheng Xu (Fellow, IEEE) is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, and the Executive Director of the China-Singapore Institute of Digital Media, Singapore.

Dr. Xu received the Best Associate Editor Award of *ACM Transactions on Multimedia Computing, Communications and Applications* in 2012 and the Best Editorial Member Award of *ACM/Springer Multimedia Systems Journal* in 2008. He served as a Program Chair of the Association for Computing Machinery (ACM) Multimedia 2009. He has served as an Associate Editor, a Guest Editor, a General Chair, a Program Chair, an Area/Track Chair, a Special Session Organizer, a Session Chair, and a Transactions on Professional Communication (TPC) Member for over 20 IEEE and ACM prestigious multimedia journals, conferences, and workshops. He is an Associate Editor of *ACM Transactions on Multimedia Computing, Communications and Applications* and the Editor-in-Chief of *ACM/Springer Multimedia Systems Journal*. He is the International Association for Pattern Recognition (IAPR) Fellow and the ACM Distinguished Scientist.



Chongyang Ma (Member, IEEE) received B.S. degree from the Fundamental Science Class (Mathematics and Physics), Tsinghua University, Beijing, China, in 2007, and the Ph.D. degree in computer science from the Institute for Advanced Study, Tsinghua University, in 2012.

He is currently a Research Lead with Kuaishou Technology, Beijing. His research interests include computer graphics and computer vision.