Towards High-Quality Photorealistic Image Style Transfer

Hong Ding, Haimin Zhang, Gang Fu, Caoqing Jiang, Fei Luo, Chunxia Xiao and Min Xu, *Member, IEEE*

***Abstract*—Preserving important textures of the content image and achieving prominent style transfer results remains a challenge in the field of image style transfer. This challenge arises from the entanglement between color and texture during the style transfer process. To address this challenge, we propose an end-to-end network that incorporates adaptive weighted least squares (AWLS) filter, iterative least squares (ILS) filter, and channel separation. Given a content image ($\mathcal{C}$) and a reference style image ($\mathcal{S}$), we begin by separating the RGB channels and utilizing ILS filter to decompose them into structure and texture layers. We then perform style transfer on the structural layers using WCT^2 (incorporating wavelet pooling and unpooling techniques for whitening and coloring transforms) in the R, G, and B channels, respectively. We address the texture distortion caused by WCT^2 with a texture enhancing (TE) module in the structural layer. Furthermore, we propose an estimating and compensating for the structure loss (ECSL) module. In the ECSL module, with the AWLS filter and the ILS filter, we estimate the texture loss caused by TE, convert the loss of the structural layer to the loss of the texture layer, and compensate for the loss in the texture layer. The final structural layer and the texture layer are merged into the channel style transfer results in the separated R, G, and B channels into the final style transfer result. Thereby, this enables a more complete texture preservation and a significant style transfer process. To evaluate our method, we utilize quantitative experiments using various metrics, including NIQE, AG, SSIM, PSNR, and a user study. The experimental results demonstrate the superiority of our approach over the previous state-of-the-art methods.**

***Index Terms*—photorealistic image style transfer, image smoothing, channel separation, texture synthesis.**

I. Introduction

Image style editing is a fundamental task in the field of image processing. One example of image style editing is artistic style transfer, which involves transferring color and texture between a photo and a reference painting image [1]–[5]. Another example is photorealistic image style transfer, which transfers the color of a reference style image to a content image while preserving the textures of the content image [6]–[14]. One major difference between artistic image style transfer and photorealistic image style transfer is that the artistic image style transfer is to keep only the basic textures of the content image $\mathcal{C}$. The result is not photorealistic, but it is artistic. The photorealistic image style transfer is to maintain the important textures of $\mathcal{C}$, the result has a photorealistic effect. However, the color and texture of images are mixed. This kind of mixing leads to the blurring between texture preservation and style transfer. Achieving good photorealistic image style transfer results depends on completely replacing the original style of $\mathcal{C}$ with the reference style of $\mathcal{S}$, while maintaining the important textures of $\mathcal{C}$. These two lines of methods have emerged as typical approaches in the image style transfer methods. The first line of approach [1], [2], [16], employs Gram matrix and a feature matrix to effectively distinguish style and texture, and produces artistic style transfer results. However, due to the inclusion of $\mathcal{C}$ content and $\mathcal{S}$ reference style in the loss function, this approach leads to the entanglement between

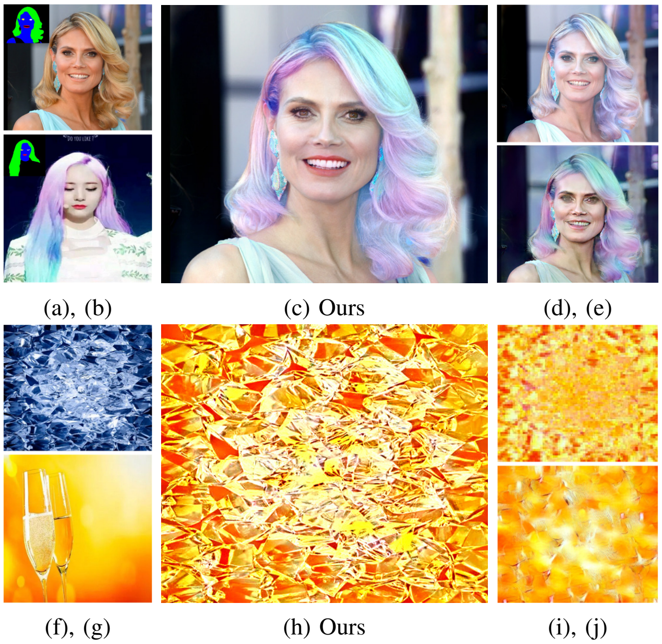


Fig. 1. Comparison with state-of-the-art methods. The first row is the face image cases, and the second row is the scene-level image cases. (a, b) Face content image ($\mathcal{C}$, top), face reference style image ($\mathcal{S}$, bottom), and their masks (at the upper left corner). (f, g) Scene-level content image ($\mathcal{C}$, top) and the scene-level reference style image ($\mathcal{S}$, bottom). (c, h) Our results of the facial image case and the scene-level image case, respectively.(d, e) Results by Ding et al. [12], and Li et al. [9], respectively. (i, j) Results by Wen et al. [15], and Hong et al. [11], respectively. Our model produces better results by utilizing hybrid filters and channel separation.

Hong Ding is with the School of Big Data and Artificial Intelligence, Guangxi University of Finance and Economics, Nanning, China and also with the School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007 Australia (Email: dhong20123@126.com)

Haimin Zhang and Min Xu (*corresponding author*) are with the School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007 Australia (Emails: haimin.zhang@uts.edu.au;min.xu@uts.edu.au)

Caoqing Jiang is with the School of Big Data and Artificial Intelligence, Guangxi University of Finance and Economics, Nanning, China (Email: jcqng@163.com)

Fei Luo and Chunxia Xiao (*corresponding author*) are with the School of Computer Science, Wuhan University, Wuhan China (Email: luofei@whu.edu.cn; cxxiao@whu.edu.cn)

Gang Fu is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (Email: xyzgfu@gmail.com)

the style and the texture, and fails to produce good image style transfer results. Subsequent methods [6], [17] have been proposed to improve upon Gatys' work [1].

The second line of significant approaches [4], [18] is the universal style transfer method, which employs texture transformation to transfer the reference styles to the content images. These universal style transfer methods embed whitening and coloring transforms (WCT) into an image reconstruction network. WCT can perform image style transfer and provide computing speed advantages. However, WCT fails to keep the textures of content images well. Various techniques have been developed to improve WCT for generating artistic or photorealistic images [8]–[12], [15], [19].

The conventional methods and their subsequent variations often compromise the preservation of texture details during color transfer, as demonstrated in Fig. 1 (e), (i), and (j). To tackle the challenge of mutual entanglement between color and texture, Ding et al. [12] proposed an adaptive filter and channel separation (AFCS) framework. The AFCS method successfully separates the structural layer and the texture layers of both the content and reference style images. It succeeds in avoiding entanglement between color transfer and texture preservation while achieving more photorealistic color transfer results. Nevertheless, it is worth noting that the textures extracted by the AWLS filter [12] in this approach tend to retain some original styles of the content image. This unclean feature extraction potentially may lead to incomplete image style transfer, as shown in Fig. 1(d) and Fig. 7(c).

This paper focuses on achieving high-quality photorealistic image style transfer. The main idea is conducting style transfer on the structural layers of $\mathcal{C}$ and $\mathcal{S}$ to avoid destroying the $\mathcal{C}$ texture extraction. We employ AWLS filter and ILS filter to extract $\mathcal{C}$ textures. This approach not only captures the essential textural details and boundary information of $\mathcal{C}$ but also effectively eliminates the original color of $\mathcal{C}$ in the texture layer. This texture extraction effect cannot be obtained only by AWLS or ILS filters.

In this paper, we design four visual ablation studies for our method, including the ILS filter, the TE (texture enhancing) module, the ECSL (estimation and compensation of the structural layer texture loss) module, and RGB channel separation, as shown in Fig 7, Fig. 9, Fig. 12, and Fig. 10. We select some representative experiments to illustrate the effectiveness of these modules. The existing quantitative evaluations primarily evaluate the relation between the content image and the output image, such as SSIM (structural similarity index) and PSNR (peak signal-to-noise ratio). They do not consider the significance of image style transfer. Consequently, we only show the effectiveness of these modules through visually significant effects. In the experimental part (Section IV), we use the visual evaluations and add quantitative evaluations in more experiments. We also add a user study to evaluate the significance of image style transfer, and obtain more comprehensive evaluation results.

This paper extends the work of Ding et al. [12]. The main differences between this paper and that of [12] are the following aspects: (1) We in this paper improve the problem of incomplete image style transfer in the work of Ding et al. [12]. (2) Our approach leverages both the AWLS filter [12] and ILS filter [20] to extract the content image textures that are independent of the content image style. (3) We present an estimation and compensation of structural layer texture loss (ECSL) module to flexibly adjust the illumination of the photorealistic image style transfer results.

We evaluate our method on a variety of images collected from the Internet and public datasets, including face images and scene-level images. The datasets used for testing include the IMDB-WIKI (Internet movie database - Wikipedia) dataset [21], the COCO (microsoft common objects in context) dataset [22], and the AFLW (annotated facial landmarks in the wild) dataset [23]. These evaluations demonstrate the effectiveness of our proposed method. The main contributions of our work can be summarized as follows.

(1) We introduce a novel approach which effectively addresses both photorealistic image style transfer and texture preservation independently, thereby reducing the entanglement of color and texture.

(2) We propose a hybrid filter based on the adaptive weighted least squares filter and the iterative least squares. We use the hybrid filter to perform image smoothing for $\mathcal{C}$ and $\mathcal{S}$, and obtain proper illumination in the photorealistic image style transfer results.

(3) We propose a TE module and an ECSL module to obtain the comprehensive preservation of $\mathcal{C}$ textures while effectively excluding the original style of $\mathcal{C}$ in the process of image style transfer.

II. Related Work

Artistic style transfer. Gatys et al. [1] used image representations extracted by CNNs optimized for artistic style transfer. Nevertheless, their method relies on a time-consuming iterative optimization process, which constrains its practicality. In contrast, Huang et al. [7] introduced a straightforward yet effective method that facilitates real-time arbitrary style transfer. Yao et al. [3] proposed multiple texture maps to capture various stroke patterns, enabling the integration of diverse strokes into different spatial regions of the output image. Nevertheless, these two approaches fail to work well when performing photorealistic image style transfer due to their limited texture-preserving capabilities for $\mathcal{C}$. To address this problem, Zhao et al. [24] introduced a method to perform artistic image style transfer using automatic semantic segmentation module produced by CNN and soft masks. Their approach eliminates more details from the content image, and obtains a more noticeable artistic style effect. Chen et al. [25] introduced a method for image sentiment transfer by using the filtered visual sentiment ontology (VSO) dataset. However, this approach has limited applicability due to its reliance on specialized datasets.

Photorealistic image style transfer. The essence of photorealistic image style transfer is color transfer. Dissimilar to artistic style transfer, which primarily preserves the basic texture of $\mathcal{C}$, photorealistic image style transfer preserves the important texture of $\mathcal{C}$ in the color transfer result. To achieve photorealistic image style transfer, many approaches have been explored in the literature. Luan et al. [6] introduced a method that combined content loss and style loss with masks. However, the tradeoff between the two losses makes it fail to produce both texture preservation and remarkable color transfer effect. Li et al. [18] proposed PhotoWCT to maximize stylization effects, yet this approach produces blurry artifacts because of the texture loss issue. Yoo et al. [8] proposed WCT^2, incorporating wavelet pooling and unpooling techniques for whitening and coloring transforms. However, this algorithm exhibited shortcomings in generating natural boundaries. Li et al. [9] introduced a data-driven fashion for transferring various styles at different levels. An et al. [10] performed style transfer through photoNet and multiple style transfer modules. Later, they introduced ArtFlow [19] to prevent content leakage in the image style transfer. Hong et al. [11] brought domain-aware style transfer networks. Wen et al. [15] proposed CAP-VSTNet (Content Affinity Preserved Versatile Style Transfer) framework for versatile style transfer. However, these methods have suffered from texture loss of $\mathcal{C}$ and results blurring. Addressing the challenge of mutual entanglement between color and texture, Ding et al. [12] introduced the AFCS framework. Nonetheless, a drawback of this approach is that the textures extracted by the AWLS filter tend to remain some style of the original $\mathcal{C}$, leading to incomplete style transfer in specific cases. On the other hand, makeup transfer approaches [26]–[30] primarily focus on transferring elements such as eye shadow, lipstick, and skin color, without addressing background alterations. These approaches fail to deal with image pairs lacking accurate dense correspondences. They also have limitations in retaining content textures. In contrast, in our work we introduce a novel method by independently performing color transfer and texture preservation with hybrid filters to deal with these issues.

Edge-preserving image smoothing. Methods based on weighted averages, as demonstrated by Dai et al. [31] and Tan et al. [32], have shown significant development over the past few decades. Farbman et al. [33] introduced an edge-preserving smoothing operator rooted in the weighted least squares (WLS) optimization framework. WLS filter can be adjusted by using multiple parameters to achieve various degrees of smoothing. However, this method lacks adaptability of parameter adjustments for different images. Barron et al. [34] presented a bilateral solver to expedite WLS smoothing process. Nevertheless, this approach is suitable only for Gaussian guidance weights and will produce artifacts in the smoothed images. Liu et al. [20] introduced an effective approach called iterative least squares (ILS) filter, which employed global optimization to achieve edge-preserving image smoothing without additional image guidance information. The method proposed in Liu et al [20] yields filtered image structural layers with more intricate details, compared to the approach of Farbman et al [33]. Fanetal et al. [35] proposed a normal decoupled learning module for image smoothing, and later introduced an unsupervised learning method [36] to improve the image smoothing. However, most of the previously mentioned methods are limited to specific applications due to their inherent fixed smoothing characteristics. Additionally, Yim et al. [37] have focused on adjusting brightness, contrast, and other settings to create smoothing effects. Therefore, in this paper we have developed an adaptive formula for selecting the parameter L in the work of Farbman et al [33]. This formula guides image smoothing specifically for the purpose of photorealistic image style transfer.

III. Methodology

In this paper, we propose a novel method to produce high-quality photorealistic image style transfer results. We introduce an adaptive weighted least squares (AWLS) filter to obtain illumination in image smoothing results, as shown in Section III-A. We use the ILS filter to extract the structural layers and texture layers of $\mathcal{C}$ and $\mathcal{S}$. Then we use the hybrid filter including the AWLS filter and the ILS filter to extract the image textures of $\mathcal{C}$, as presented in Section III-B. This is the key step to extract the important textures of $\mathcal{C}$ without extracting the original color of $\mathcal{C}$, as shown in the formation process of T_j ($j \in \{R, G, B\}$ channel) in Fig. 2. Specifically, we perform color transfer in the two structural layers from the ILS filter, as shown in Section III-C. Then we utilize the AWLS filter to perform texture enhancing (TE) to repair the unnatural border of the output from WCT^2, as presented in Section III-C3, and carry out estimation and compensation of the structure layer texture loss (ECSL) via the hybrid filters, as introduced in Section III-D. In the ECSL module, we estimate the texture loss of the structural layer, caused by TE, and compensate for the texture loss in the texture layer. Finally, we obtain the photorealistic image style transfer result O by merging the outputs O_j ($j \in \{R, G, B\}$ channel), as presented in Section III-E.

A. Adaptive weighted least squares filter (AWLS)

We use the AWLS filter [12] in the TE module and the ECSL module in Fig. 2. Weighted least squares (WLS) [33] aims to generate a new image denoted as u from an input image f. u is as close as possible to f, and, at the same time, is as smooth as possible every-where, except across significant gradients in f. Mathematically, this can be expressed as the energy minimization of the following objective function.

$$\sum_s \left((u_s - f_s)^2 + \lambda (a_{x,s}(f) (\tfrac{\partial u}{\partial x})_s^2 + a_{y,s}(f) (\tfrac{\partial u}{\partial y})_s^2) \right), \quad (1)$$

where s represents the spatial location of a pixel. The primary objective of the data term $(u_s - f_s)^2$ is to minimize the difference between u and f, while the second (regularization) term focuses on achieving smoothness by minimizing the partial derivatives of u. λ is responsible for the balance between the two terms. The smoothness weights $a_{x,s}(f)$ and $a_{y,s}(f)$ are defined as follows.

$$\begin{aligned} a_{x,s}(f) &= \left(|\tfrac{\partial l}{\partial x}(s)|^{\alpha'} + \varepsilon' \right)^{-1}, \\ a_{y,s}(f) &= \left(|\tfrac{\partial l}{\partial y}(s)|^{\alpha'} + \varepsilon' \right)^{-1}, \end{aligned} \quad (2)$$

where l is the log-luminance of the input image f, α' is a parameter between 1.2 and 2.0, and it determines the sensitivity to the gradients of f, while ε' is a small constant

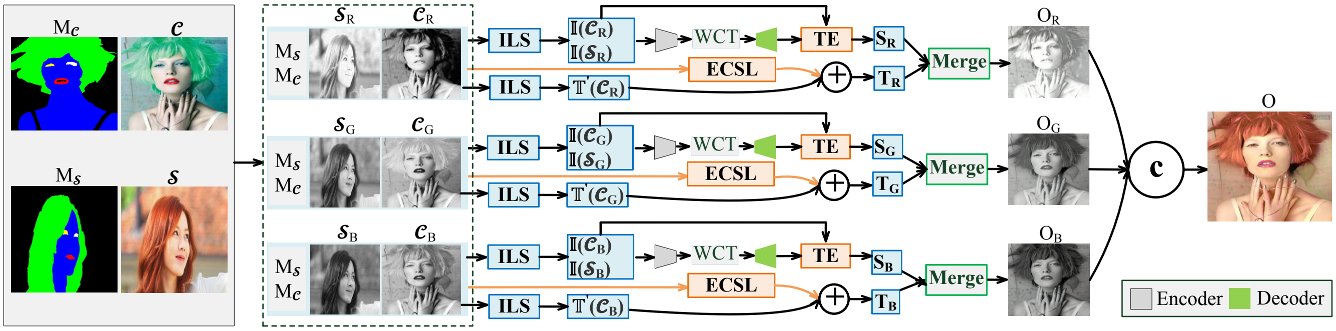

$M_{\mathcal{C}}$
$\mathcal{C}$
$M_{\mathcal{S}}$
$\mathcal{S}$
$\mathcal{S}_R$ $\mathcal{C}_R$
$M_{\mathcal{S}}$ $M_{\mathcal{C}}$
ILS
$\mathbb{I}(\mathcal{C}_R)$ $\mathbb{I}(\mathcal{S}_R)$
WCT
TE
ECSL
$\mathbb{T}'(\mathcal{C}_R)$
S_R T_R
Merge
O_R
$\mathcal{S}_G$ $\mathcal{C}_G$
$\mathbb{I}(\mathcal{C}_G)$ $\mathbb{I}(\mathcal{S}_G)$
$\mathbb{T}'(\mathcal{C}_G)$
S_G T_G
O_G
$\mathcal{S}_B$ $\mathcal{C}_B$
$\mathbb{I}(\mathcal{C}_B)$ $\mathbb{I}(\mathcal{S}_B)$
$\mathbb{T}'(\mathcal{C}_B)$
S_B T_B
O_B
c
O
Encoder
Decoder

Fig. 2. Overview of our framework. First, our model takes a content image $\mathcal{C}$, a reference style image $\mathcal{S}$, and their corresponding mask maps $M_{\mathcal{C}}$ and $M_{\mathcal{S}}$ as inputs. The ILS filter is employed to extract the R, G and B structural layers of $\mathcal{C}$ and $\mathcal{S}$ ($\mathbb{I}(\mathcal{C}_j)$ and $\mathbb{I}(\mathcal{S}_j)$, $j \in \{R, G, B\}$ channel). Then we input $\mathbb{I}(\mathcal{C}_j)$, $\mathbb{I}(\mathcal{S}_j)$, $M_{\mathcal{C}}$, and $M_{\mathcal{S}}$ into the WCT^2 model [8] to transfer the style of the three structural layers, and use TE module to enhance the texture for the WCT^2 output to get $\mathbf{S}_j$. We use the ECSL module to estimate and compensate for the texture loss caused by TE. Next, for texture preservation, we assign weights to the texture layers extracted by ILS filter ($\mathbb{T}'(\mathcal{C}_j)$) and the ECSL output, to obtain the final texture $\mathbf{T}_j$. Finally, we obtain the photorealistic color transfer result O_j by merging $\mathbf{S}_j$ and $\mathbf{T}_j$.

that prevents division by zero in areas where f is constant. In fact, we can utilize other matrices instead of the parameter l.

If the function $M(f)$ presents a source image for the affinity matrix, which has the same dimension as the input image f. The solution for Eq. 1 is

$$(I + \lambda M(f))u = f, \tag{3}$$

where the default value $M(f)$ is $log(f)$, I is the unit matrix.

The research [12] shows that when we smooth $\mathcal{C}$ and $\mathcal{S}$ with the WLS filter, the parameter M of the WLS filter can affect the illumination of image smoothing and style transfer results, as shown in Fig. 3 and Fig. 4. At present, we fail to find existing methods designed to smooth a pair of images ($\mathcal{C}$ and $\mathcal{S}$) at the same time in the photorealistic image style transfer process. To address this limitation, we introduce an adaptive weighted least squares (AWLS) filter to adaptively fine-tune the key parameter $\mathscr{L}(f)$ of the WLS filter for $\mathcal{C}$ and $\mathcal{S}$ to obtain style transfer result with proper illumination. When we employ the illumination channel, L channel from LAB color space, to perform image smoothing with AWLS, the result suffers from drawbacks, such as frequently blurring the boundaries. Therefore, we consider both $log(L(f))$ and $log(f)$ to guide the image smoothing, where $L(\cdot)$ denotes the luminance of an image.

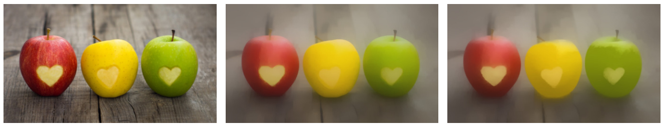
Both $\mathcal{S}$ and $\mathcal{C}$ are inputed into Eq. (3) as f. WLS filtering is performed channel-by-channel in RGB spaces. When f is $\mathcal{C}$, we set $\mathbb{L}(\mathcal{C}_j)$ as the weighted sum of $L(\mathcal{C})$ and $\mathcal{C}_j$:

$$\mathbb{L}(\mathcal{C}_j) = \begin{cases} log(L(\mathcal{C})), \Delta L = 1 \\ log(\mathcal{C}_j), \Delta L < 0.5 \\ \alpha \times log(L(\mathcal{C})) + (1-\alpha) \times log(\mathcal{C}_j), \\ 0.5 \leq \Delta L < 1, \end{cases} \tag{4}$$

where $j \in \{R, G, B\}$, and

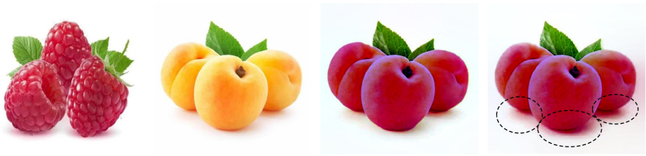
$$\alpha = \beta \times (\tfrac{\Delta L}{0.5} - 1). \tag{5}$$

Here β is a trigger whose value is 0 or 1. When $\Delta L \geq 0.5, \beta$ =1, and when $\Delta L < 0.5$, β=0. ΔL =$|mean(L(\mathcal{C})) - mean(L(\mathcal{S}))|$. $mean(\bullet)$ denotes the mean value of all elements in a matrix. The constant 0.5 is a domain value which is used to control the selection between $log(L(\mathcal{C}))$ and $log(\mathcal{C}_j)$. This constant is set by experimental experience. Furthermore, because the maximum of ΔL is 1, according to Eq. 5, the maximum of α is 1.



(a) Input (b) Without L (c) With L

Fig. 3. Influence of the L channel in AWLS filter. (a) Input image. Smoothing an image with its L channel, we can obtain the changed illumination information, as in the top of the fruits of (c). In (b), without the L channel for guided smoothing, the result is mainly to remove image textures, with less illumination variations than in (c).



(a)$\mathcal{S}$ (b) $\mathcal{C}$ (c) Without L (d) With L

Fig. 4. Effect of ΔL in Eq. 4. (a) Reference style image. (b) Content image. When the luminance L of $\mathcal{C}$ and $\mathcal{S}$, are similar ($\Delta L < 0.5$), we obtain better contours without L than with L (see the fruits' shadows circled in black).

The smoothing results of the three channels are merged into the final smoothing results. When f is R, we compute $L(\mathcal{S})$ like $L(\mathcal{C})$. We leverage $L(\mathcal{C})$ to guide the image smoothing, which preserves not only the image color, but also the brightness variation information. We show the influence of L in Eq. 4 in Fig. 3 and Fig. 4 using the AFCS method [12]. This influence will exist as long as we use the WLS filter for photorealistic image style transfer.

B. Hybrid Filter Composed of ILS and AWLS filters

In this section, we first review the ILS filter, then compare it with the AWLS filter, and finally mix the two filters to avoid the problem of incomplete transfer, as shown in Fig. 1(d).

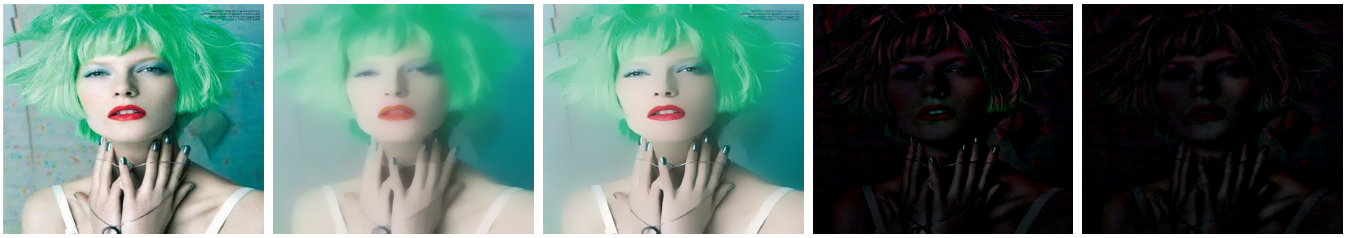

(a) Input (b) AWLS-s (c) ILS-s (d) AWLS-f (e) ILS-f

Fig. 5. Image smoothing comparison. (a) Input image. (b, d) Structure and texture layers extracted by AWLS filter, respectively. (c, e) Structure and texture layers extracted by ILS filter, respectively.

1) ILS: The ILS filter [20] is from the minimization of the following objective function.

$$E(u,f)=\sum_{s}\left((u_s-f_s)^2+\lambda\sum_{*\in\{x,y\}}\phi_p(\nabla u_{*,s})\right), \quad (6)$$

where f is an input image, u is the smoothed image, s denotes a pixel location, and $\nabla u_*(*\in\{x,y\})$ represents the gradient of u along horizontal and vertical directions, respectively. The penalty function $\phi_p(\cdot)$ is defined as

$$\phi_p(x)=(x^2+\varepsilon)^{\frac{p}{2}}. \quad (7)$$

The norm power p is typically set as $0<p\leq 1$ for edge-preserving smoothing, and ε = 0.0001. The solution for Eq. 6 is

$$u^{n+1}=\mathcal{F}^{-1}\left(\frac{\mathcal{F}(f)+\frac{\lambda}{2}\overline{\mathcal{F}(\nabla_x)}\cdot\mathcal{F}(\mu_x^n)+\overline{\mathcal{F}(\nabla_y)}\cdot\mathcal{F}(\mu_y^n)}{\mathcal{F}(1)+\frac{c}{2}\cdot\lambda(\overline{\mathcal{F}(\nabla_x)}\cdot\mathcal{F}(\nabla_x)+\overline{\mathcal{F}(\nabla_y)}\cdot\mathcal{F}(\nabla_y)}\right), \quad (8)$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ are the fast Fourier transform and inverse fast Fourier transform operators, respectively. $\overline{\mathcal{F}(\cdot)}$ denotes the complex conjugate of $\mathcal{F}(\cdot)$, $c=p\varepsilon^{\frac{p}{2}-1}$. The addition, multiplication and division are all element-wise operations.

2) Comparison between ILS and AWLS filters: In Eq. 6, the second term of the loss is 1NF (normal form) in the case of ε = 0 and p= 1. In Eq. 1, the second term of loss is 2NF in the case of no weights applied. Therefore, Eq. 6 preserves more edge details because 1NF is a looser constraint than 2NF. [38]

Therefore, the texture layer from the ILS filter keeps relatively fewer textures and nearly no color information. Even smoothing an image with bright colors, the ILS filter can extract a texture layer with less color information, as shown in Fig. 5. Therefore, using the ILS filter to smooth the image can reduce the problem of incomplete style transfer, as shown in Fig. 7. More visual comparisons between the AWLS filter and the ILS filter are shown in Fig. 11.

3) Hybrid filter composed of ILS filter and AWLS filter: When $\mathcal{C}$ and $\mathcal{S}$ are input, the output W of WCT^2 [8] model will show artificial boundaries that need to be repaired. The texture layer of $\mathcal{C}$ from the ILS filter keeps little boundary information of $\mathcal{C}$, and is not suitable for repairing artificial boundaries, such as color overflow and unnatural boundaries. In contrast, the texture layer of $\mathcal{C}$ from the AWLS filter contains more textures, including boundary information, and is suitable for image boundary repair. However, this approach also introduces some original colors of $\mathcal{C}$ into the repaired results. Therefore, in this paper we propose a hybrid filter composed of AWLS filter and ILS filter. The texture layer from this hybrid filter keep little the original image colors, and can repair the unnatural boundaries of W.

During the style transfer process, we use the ILS filter to extract the structural layers from $\mathcal{S}$ and $\mathcal{C}$. The WCT^2 [8] model is then utilized to transfer the style of $\mathcal{S}$ structural layer to $\mathcal{C}$ structural layer. Subsequently, we use the AWLS filter to repair the artificial boundaries caused by WCT^2. However, it should be noted that due to the difference in principle and the effect between the AWLS filter and the ILS filter, some useful textures in the structural layer may be lost in the repair process. Therefore, in the texture preservation stage, the ILS filter is employed to extract the texture layer and the AWLS filter is used to compensate for the lost useful texture in structural layer repair.

In Fig. 6, we illustrate the sequential steps for obtaining the j-channel final textures in the photorealistic image style transfer. Specifically, to initiate the process, we employ the WCT^2 method to transfer the style of $\mathcal{S}$ to $\mathcal{C}$ to generate W_j; we repair W_j using TE module (see Section III-C3) to obtain W'_j; we utilize AWLS filter to extract $\mathbb{W}(W'_j)$, the structural layer of W'_j; we obtain $\mathbb{T}(W'_j)$ (the texture layer of W'_j) by using $W'_j-\mathbb{W}(W'_j)$. Finally, we perform weighted fusion of $\mathbb{T}(W'_j)$ and $\mathbb{T}(\mathcal{C}_j)$ to obtain the final texture $\mathbf{T}_j$ of the j channel. The procedures are shown in Fig. 6. The orange dotted rectangle is the ECSL module. The theoretical reason of ECSL is illustrated in Section III-D. The photorealistic image style transfer comparisons between the AFCS method [12] and ours are shown in Fig. 7.

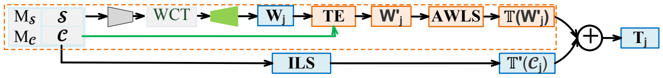
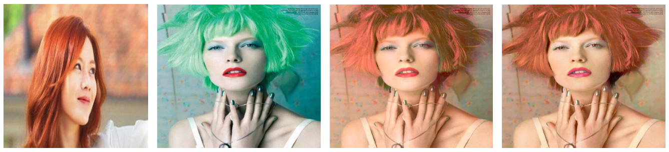


Fig. 6. Overview of our texture preserving. The dotted rectangle in orange is the ECSL module.

C. Channel-separated style transfer

1) Encoder and decoder architecture: We employ encoder and decoder architecture in WCT^2 [8] for color transfer. The encoder utilizes the Image Net-pretrained VGG-19 network, incorporating Haar wavelet pooling [8] from the conv1_1 layer to the conv4_1 layer. The decoder mirrors the structure of



(a) $\mathcal{S}$ (b) $\mathcal{C}$ (c) without ILS (d) with ILS

Fig. 7. Ablation study for ILS filter. (a) Reference style image. (b) Content image. (c) Style transfer result without the ILS filter. (d) Style transfer result with the ILS filter. By adding the ILS filter in the image style transfer model, we can get the final texture layer without the original color of $\mathcal{C}$, and obtain a better photorealistic image style transfer result.

the encoder. With a noteworthy texture enhancing module, we augment the decoder with texture enhancing (TE) module placed behind the convolutional layer in each decoder layer. This augmentation addresses the issue of unnatural boundary effects as illustrated in Fig. 8.

2) Channel separation photorealistic image style transfer [12]: To avoid the entanglement among the R, G, and B channels, we perform style transfer in each channel (R, G, and B) in the structural layers of $\mathcal{C}$ and $\mathcal{S}$, respectively. Then we achieve style transfer result of the structural layer $\mathbf{S}_j$, where $j \in \{R, G, B\}$ channel, as illustrated in Fig. 2. While the concept of separating RGB channels has been employed in certain color-related tasks [39], [40], as far as our knowledge extends, we pioneered the application of a deep learning network based on color channel separation in the realm of photorealistic image style transfer. This approach is inspired by the ″divide-and-grow″ concept. We employ semantic segmentation to derive masks $M_{\mathcal{C}}$ and $M_{\mathcal{S}}$ for $\mathcal{C}$ and $\mathcal{S}$, respectively. For each R, G, or B channel, we perform color transfer by feeding the R, G, and B components of $\mathbb{I}(\mathcal{C})$ and $\mathbb{I}(\mathcal{S})$, along with their respective masks, into the encoder-decoder modules (as depicted in Fig. 2). The fundamental principle guiding this color transfer process is rooted in the principles of WCT [1].

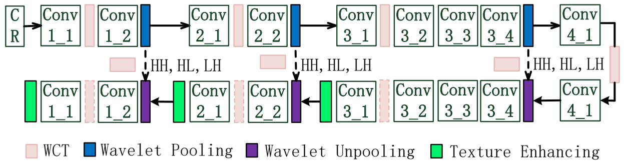


Fig. 8. The structure of our used encoder-decoder network. We add a texture enhancing operation after each convolution layer in the decoder [8].

When we perform style transfer from $\mathcal{S}$ to $\mathcal{C}$, we extract the texture $t^{\mathcal{C}}$ of $\mathcal{C}$ from the decoder of WCT:

$$t'^{\mathcal{C}} = E^{\mathcal{C}}(D^{\mathcal{C}})^{-1/2}(E^{\mathcal{C}})^T t^{\mathcal{C}}, \tag{9}$$

$D^{\mathcal{C}}$ is a diagonal matrix with the eigenvalues of the covariance matrix $t'^{\mathcal{C}}(t'^{\mathcal{C}})^T \in \mathcal{R}^{Ch\times Ch}$, and $E^{\mathcal{C}}$ is the corresponding orthogonal matrix of eigenvectors, satisfying $t'^{\mathcal{C}}(t'^{\mathcal{C}})^T = E^{\mathcal{C}}D^{\mathcal{C}}(E^{\mathcal{C}})^T$. $t^{\mathcal{C}}$ is the vectorized VGG texture of $\mathcal{C}$. Ch is the number of channels.

We transfer the color from $\mathcal{S}$ to $\mathcal{C}$ by

$$t''^{\mathcal{C}} = E^{\mathcal{S}}(D^{\mathcal{S}})^{-1/2}(E^{\mathcal{S}})^T t'^{\mathcal{C}}, \tag{10}$$

where $t''^{\mathcal{C}}$ is the color transfer result, $D^{\mathcal{S}}$ is a diagonal matrix with the eigenvalues of the covariance matrix $t'^{\mathcal{S}}(t'^{\mathcal{S}})^T \in \mathcal{R}^{Ch\times Ch}$, and $E^{\mathcal{S}}$ is the corresponding orthogonal matrix of eigenvectors. $t''^{\mathcal{C}} = t''^{\mathcal{C}} + m$, where m is the mean vector of $t'^{\mathcal{C}}$. We invert $t''^{\mathcal{C}}$ to the decoder to obtain the color transformation result.

When we perform style transfer from $\mathbb{I}(\mathcal{S}_j)$ to $\mathbb{I}(\mathcal{C}_j)$ (shown in Fig. 2), we can get the color transfer result $\mathbf{S'}_j$ in the RGB channel by replacing $\mathcal{C}$ with $\mathbb{I}(\mathcal{C}_j)$, replacing $\mathcal{S}$ with $\mathbb{I}(\mathcal{S}_j)$ in Eq. 9 and Eq. 10, where $j \in \{R, G, B\}$ channel.

3) Texture Enhancing (TE) with AWLS filter: WCT^2 [8] has unnatural boundary problems (see Fig. 9 (c) and (e)) caused by the biased style transfer claimed by [15], [19]. Our method uses filters to process the textures of the input image separately. We only perform color transer at the structural layer. We use the TE module to enhance the color transfer results of WCT^2 in the structural layer. These two methods enable us to improve the biased style transfer problem.

We repair the WCT^2 output (W) by replacing the AWLS texture layer of W with that of $\mathcal{C}$. We extract the structural layer $\mathbb{W}(W)$ of W using AWLS filter. Then, we extract the texture layer $\mathbb{T}(\mathcal{C})$ of $\mathcal{C}$ using AWLS filter by $\mathcal{C} - \mathbb{W}(\mathcal{C})$. Finally, we fuse $\mathbb{W}(W)$ and $\mathbb{T}(\mathcal{C})$ to generate W', the texture enhancing result of W. The texture enhancing can be expressed as

$$W' = \mathbb{W}(W) + \mathbb{T}(\mathcal{C}) = \mathbb{W}(W) + \mathcal{C} - \mathbb{W}(\mathcal{C}), \tag{11}$$

where $\mathbb{W}(\cdot)$ denotes the AWLS filter operator, $\mathbb{W}(z)$ and $\mathbb{T}(z)$ are the structure and texture layers of an variable z using AWLS filter, respectively.

We can set the weights of $\mathcal{C}$ and $\mathbb{W}(\mathcal{C})$ to get slightly different results of $\mathbb{T}(\mathcal{C})$:

$$\mathbb{T}(\mathcal{C}) = k_1 \times \mathcal{C} - k_1' \times \mathbb{W}(\mathcal{C}), \tag{12}$$

where k_1 and k_1' are the weights of $\mathcal{C}$ and $\mathbb{W}(\mathcal{C})$, respectively. We also use the AWLS filter to obtain the texture layer $\mathbb{T}(W)$ of W. According to Eq. 12, we can get $\mathbb{T}(W)$ as follows.

$$\mathbb{T}(W) = k_1 \times W - k_1' \times \mathbb{W}(W). \tag{13}$$

We use the ILS filter to obtain $\mathcal{C}$ texture layer which has no the original color of $\mathcal{C}$:

$$\mathbb{T}'(\mathcal{C}) = k_2 \times \mathcal{C} - k_2' \times \mathbb{I}(\mathcal{C}), \tag{14}$$

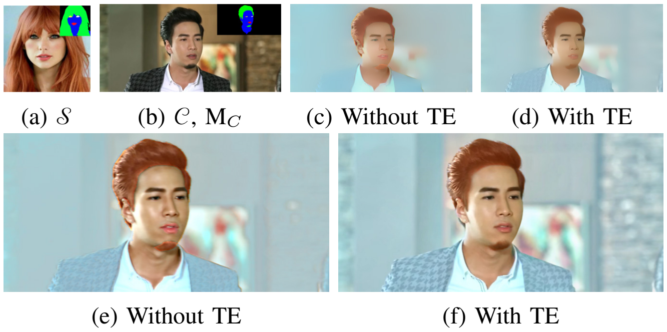
where $\mathbb{I}(\cdot)$ denotes ILS operator, and $\mathbb{I}(z)$ and $\mathbb{T}'(z)$ are the structure and texture layers of z using ILS filter, respectively. k_2 and k_2' are the weights of $\mathcal{C}$ and $\mathbb{I}(\mathcal{C})$, respectively. We typically set $k_2 = k_2' = 1$. We also use the ILS filter to obtain the texture layer $\mathbb{T}'(W)$ of W. According to Eq. 14, we can get $\mathbb{T}'(W)$ as follows.

$$\mathbb{T}'(W) = k_2 \times W - k_2' \times \mathbb{I}(W). \tag{15}$$

Hence, when we use TE to repair W_j (WCT^2 output W in j channel, as shown in Fig. 6), Eq. 11 can be rewritten as

$$\mathbf{S}_j = \mathbb{W}(\mathbb{I}(W_j)) + \mathbb{I}(\mathcal{C}_j) - \mathbb{W}(\mathbb{I}(\mathcal{C}_j)), \tag{16}$$

where $\mathbf{S}_j$ is the TE repair result of the structural layer in the WCT^2 output in the $j (j \in \{R, G, B\})$ channels. $\mathcal{C}_j$ is the part


(a) $\mathscr{S}$ (b) $\mathscr{C}$, M_{C} (c) Without TE (d) With TE

(e) Without TE (f) With TE

Fig. 9. Ablation study for Texture Enhancing (TE). (a) $\mathscr{S}$, $\mathrm{M}_{\mathscr{S}}$ (at the upper right corner). (b) $\mathscr{C}$, $\mathrm{M}_{\mathscr{C}}$ (at the upper right corner). (c, d) Our style transfer results for the structural layers without and with TE, respectively. (e, f) Our final style transfer results without and with TE, respectively.

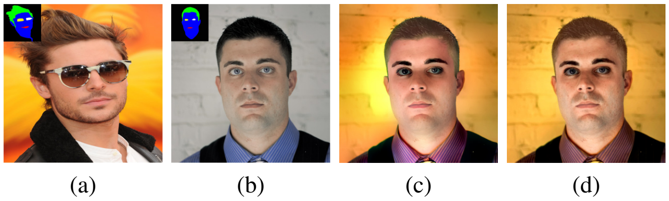

(a) (b) (c) (d)

Fig. 10. Ablation study for our method without and with RGB channel separation. (a) (b) $\mathscr{S}$, $\mathscr{C}$, and their masks (at the upper left conner), (c) Result of our method without RGB channel separation. (d) Result of our method with RGB channel separation.

of $\mathscr{C}$ in j channel. $\mathbb{W}(\mathbb{I}(W_j))$ is AWLS structural layer of the ILS smoothing result of W_j. $\mathbb{I}(\mathscr{C}_j)$ is the structural layer $\mathscr{C}_j$ from ILS filter, and $\mathbb{W}(\mathbb{I}(\mathscr{C}_j))$ is the AWLS structural layer of the ILS structural layer of $\mathscr{C}_j$. Fig. 9(d) shows the visual result of $\mathbf{S}$. With the TE module, the results have more natural textures.

The ablation experiment of channel separation is shown in Fig. 10. Image style transfer based on channel separation can obtain more uniform image style transfer results.

D. Estimation and compensation of the structural layer texture loss (ECSL)

During the process of color transfer in structural layer, repairing W_j from WCT^2 with TE may lead to the loss of textures in the structural layer. We estimate the texture loss, convert the loss of the structural layer to the loss of the texture layer, and compensate for the loss in the texture layer.

For the same image, we use WLS filter and ILS filter to decouple it, and get different structural layer and texture layer. However, the coupling result of structural layer and texture layer obtained by WLS should be approximately equal to that obtained by ILS filter. Hence, when we utilize AWLS filter and ILS filter to smooth W, respectively, according to Eq. 12 and Eq. 14, we obtain this relation.

$$\mathbb{W}(W) + \mathbb{T}(W) \approx \mathbb{I}(W) + \mathbb{T}'(W). \tag{17}$$

The three filters shown on our paper are ILS filter, AWLS filter with L, and AWLS filter without L, which are shown in Section III-B. We analyze the main differences in these filters through two representative examples, illustrated in Fig. 11. In the first row, the images consist of color blocks with fewer textures, and the second row is characterized by more textures and fewer color blocks. In the first row, the smoothing of the AWLS filter with L (denoted as AWLS$_1$) is based on the gradient of the image illumination, thereby yielding results that accurately represent illumination variations. However, illumination is only part of the image, so this approach may introduce blurriness at boundaries, as shown in (b). Because the texture layer equals to the input minus the structural layer, the texture layer from this method retains boundary information and some of the original input's color, as shown in (c).

In contrast, the AWLS filter without L (denoted as AWLS$_2$) employs the image itself for gradient, resulting in outcomes unaffected by brightness variations and offering superior edge preservation. Nevertheless, it has some shortcomings in capturing the rich information of illumination change compared to AWLS$_1$, as indicated by (d). Furthermore, the texture extracted by AWLS$_2$ is fewer than that of AWLS$_1$, as illustrated by (e). In the second row, the input image is mainly textured with fewer color blocks. The smoothing results of AWLS$_1$ and AWLS$_2$ are very similar to the texture extraction results due to the weakening influence of brightness. Both methods can preserve most of the details of the image.

In the case of the ILS filter, it employs the image itself for gradient while incorporating an edge preservation factor p (shown in Eq. 7). Therefore, the ILS filter produces smoothing results with more edge preservation compared to AWLS$_1$ and AWLS$_2$, as shown in (f). However, the texture retaining in the texture layer from the ILS filter is considerably less than that of AWLS$_1$ and AWLS$_2$, as shown in (g). In the texture layer, the ILS filter removes more color of the original image, and only purer texture details are available. Hence, the ILS filter has better texture extraction ability than AWLS$_1$ and AWLS$_2$.

The ILS filter aims to find the gradient of image pixel values and set the edge-preserving factor, resulting in a superior smoothing effect for preserving image texture of edges. On the other hand, AWLS$_1$ relies on the image brightness information to better preserve the overall brightness change of the image. Consequently, we can obtain the following relations.

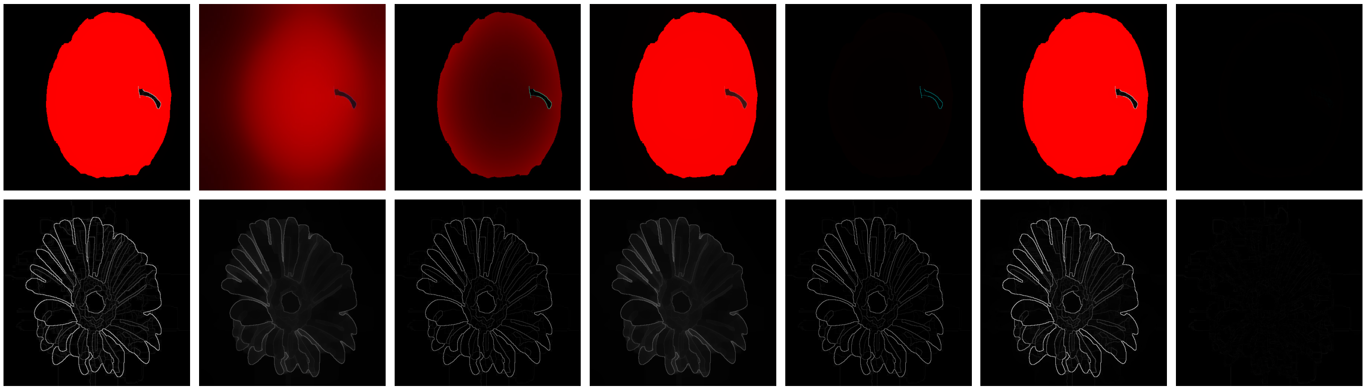
$$\begin{aligned} &\mathbb{W}(\mathbb{I}(W)) \approx \mathbb{W}(W), \\ &\mathbb{W}(\mathbb{I}(\mathscr{C})) \approx \mathbb{W}(\mathscr{C}). \end{aligned} \tag{18}$$

We use the difference between the WCT^2 output and the repaired result of the WCT^2 output of TE module as the texture loss caused by TE module. When $\mathbb{I}(W_j)$ is the output of WCT^2 of the structural layer of j channel in Fig. 2, $\mathbf{S}_j$ is the repaired result of the WCT^2. According to Eq. 16, we have the texture loss in the structural layer repairing:

$$\begin{aligned} &\delta = \mathbb{I}(W_j) - \mathbf{S}_j \\ &= \mathbb{I}(W_j) - \mathbb{W}(\mathbb{I}(W_j)) - \mathbb{I}(\mathscr{C}_j) + \mathbb{W}(\mathbb{I}(\mathscr{C}_j)) \\ &\approx \mathbb{I}(W_j) - \mathbb{W}(W_j) - \mathbb{I}(\mathscr{C}_j) + \mathbb{W}(\mathscr{C}_j) \\ &\approx W_j - \mathbb{T}'(W_j) - \mathbb{W}(W_j) - (\mathscr{C}_j - \mathbb{T}'(\mathscr{C}_j)) + \mathbb{W}(\mathscr{C}_j) \\ &\approx \mathbb{T}(W_j) - \mathbb{T}'(W_j) - \mathbb{T}(\mathscr{C}_j) + \mathbb{T}'(\mathscr{C}_j), \end{aligned} \tag{19}$$

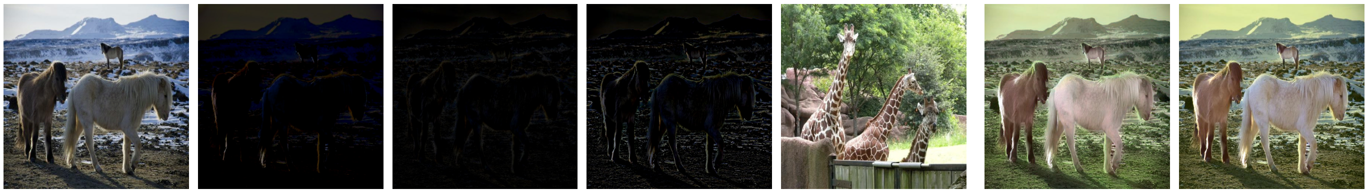
where δ is the texture loss in the structural layer repairing.

The formula above has 4 texture layer variables, which can be simplified. Since Eq. 11 also contains the two variables w



(a) Input (b) AWLS$_1$-s (c) AWLS$_1$-f (d) AWLS$_2$-s (e) AWLS$_2$-f (f) ILS-s (g) ILS-f

Fig. 11. Comparison of the three image filters. (a) Input. (b, c) Structure and texture layers produced by AWLS$_1$ filter, respectively. (d, e) Structure and texture layers produced by AWLS$_2$ filter, respectively. (f, g) Structure and texture layers produced by ILS filter, respectively.



(a) $\mathcal{C}$ (b) WCT^2 (c) ILS (d) ILS + ECSL (e)$\mathcal{S}$ (f) ILS (g) ILS + ECSL

Fig. 12. Ablation study for ECSL module. (a) Content image. (b)-(d) Extracted texture maps by WCT^2, ILS, and ILS+ECSL, respectively. (e) Reference style image. (f, g) Style transfer results produced by our method without and with the ECSL module, respectively.

and c, we convert the structural layers in Eq. 11 into texture layers. Hence, we take $\mathbb{W}(W) = W - \mathbb{T}(W)$ into Eq. 11 to get

$$W' = \mathbb{W}(W) + \mathbb{T}(\mathcal{C}) = W - \mathbb{T}(W) + \mathbb{T}(\mathcal{C}). \tag{20}$$

To get a result closer to the Eq. 19, we extract the AWLS texture from W'_j. According to Eq. 20 we get:

$$\begin{aligned} \mathbb{T}(W'_j) &= \mathbb{T}(W_j - \mathbb{T}(W_j) + \mathbb{T}(\mathcal{C}_j)) \\ &\approx \mathbb{T}(W_j) - \mathbb{T}(\mathbb{T}(W_j)) + \mathbb{T}(\mathbb{T}(\mathcal{C}_j)). \end{aligned} \tag{21}$$

We represent the texture loss δ with the weighted sum of $K_1 \times \mathbb{T}(W'_j)$ and $K'_1 \times \mathbb{T}(\mathcal{C}_j)$ approximately. We regard the difference between δ and the weighted sum as the difference between the unnatural boundary texture and the natural boundary texture. Hence, we have the final texture layer of j channel $\mathbf{T}_j$ with another $\mathbb{T}(\mathcal{C}_j)$ in Fig. 6.

$$\mathbf{T}_j = K_1 \times \mathbb{T}(W'_j) + K'_1 \times \mathbb{T}'(\mathcal{C}_j), \tag{22}$$

where K_1 and K'_1 are the weights of $\mathbb{T}(W'_j)$ and $\mathbb{T}(\mathcal{C}_j)$, respectively. Fig. 12 shows the comparison of texture extraction and style transfer using the ILS filter and the ECSL module.

The brightness information from the AWLS filter is stored in both the structure layer and the texture layer. Combining the color transfer of the structural layer with AWLS filter and ILS filter can affect the significance of brightness information preservation. With the AWLS filter parameters unchanged, we can flexibly adjust the brightness effect of the final style transfer result by modifying the weights of AWLS and ILS texture in Eq. 22. However, the details in the ILS texture are limited, as shown in Fig. 12 (c) and (f). According to the experimental experience, the weight of the ILS texture layer should not be set too high. It should be generally no more than 0.4. The AWLS texture layer weight should not be less than 0.6. Fig. 13 illustrates the results of image style transfer under various parameters.

We note that both α in Eq. 4 and K_1, K'_1 in Eq. 22 influence the brightness of the image style transfer result. However, if α is set to considerably high, it may lead to blurring at the boundaries of the final result. When α falls into a suitable value range in Eq. 4, increasing K'_1 over 0.5 in Eq. 22 does not damage the image quality. However, it will affect the overall brightness of the final result.

E. Mergence of structure and texture layers

We achieve the photorealistic image style transfer result of j channel O_j ($j \in \{R, G, B\}$ channel) by merging $\mathbf{S}_j$ and $\mathbf{T}_j$.

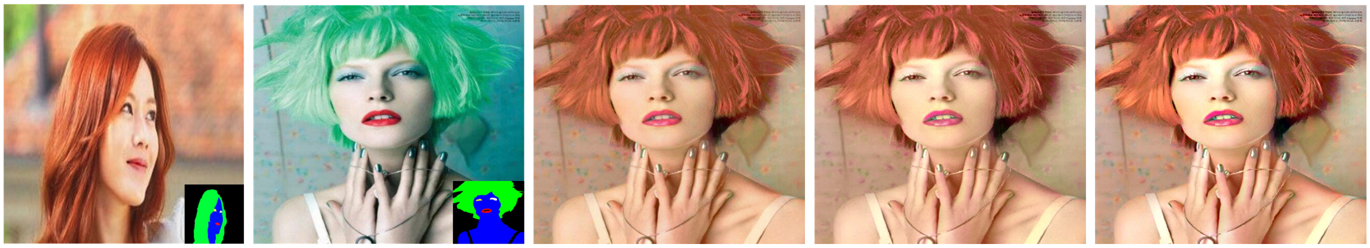
$$O_j = \gamma_1 \mathbf{S}_j + \gamma_2 \mathbf{T}_j, \tag{23}$$

where γ_1 and γ_2 are the weights of $\mathbf{S}_j$ and $\mathbf{T}_j$. We replace $\mathbf{T}_j$ in Eq. 23 with Eq. 22 to obtain O_j as follows.

$$O_j = \gamma_1 \mathbf{S}_j + \gamma_2 (K_1 \times \mathbb{T}(W'_j) + K'_1 \times \mathbb{T}'(\mathcal{C}_j)) \tag{24}$$

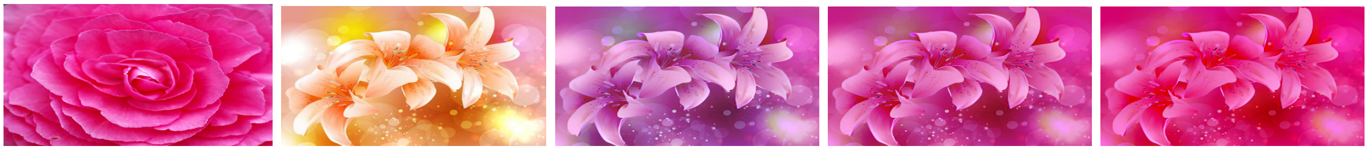
In Eq. 12, when we use AWLS filter to smooth W'_j, we get $\mathbb{T}(W'_j) = k_1 W'_j - k'_1 \mathbb{W}(W'_j)$. Hence, we obtain

$$\begin{aligned} O_j &= \gamma_1 \mathbf{S}_j + \gamma_2 [K_1 (k_1 W'_j - k'_1 \mathbb{W}(W'_j)) + K'_1 \mathbb{T}'(\mathcal{C}_j)] \\ &= \gamma_1 \mathbf{S}_j + \gamma_2 K_1 (k_1 W_j^{'} - k'_1 \mathbb{W}(W'_j)) + \gamma_2 K'_1 \mathbb{T}(\mathcal{C}_j) \\ &= \gamma_1 \mathbf{S}_j + \gamma_2 K_1 k_1 W'_j - \gamma_2 K_1 k'_1 \mathbb{W}(W'_j) + \gamma_2 K'_1 \mathbb{T}'(\mathcal{C}_j). \end{aligned} \tag{25}$$



(a) $\mathcal{S}$ (b) $\mathcal{C}$ (c) K_1 =0.4, K_1' =0.6 (d) K_1 =0.3, K_1' =0.7 (e) K_1 =0.1, K_1' =0.9

Fig. 13. Style transfer results with varying parameter K_1 and K_1' in Eq. 22. (a) and (b) Reference style image, content image, and their masks (at the lower right corner). (c)-(e) Style transfer results with varying K_1 and K_1'. We obtain better luminance information with larger K_1' (see the nose and forehead). The weights K_1 and K_1' are shown under the images.



(a) $\mathcal{C}$ (b) $\mathcal{S}$ (c) γ_1 =0.4, $-k_1'$ =-0.4 (d) γ_1 =0.7, $-k_1'$ =-0.7 (e) γ_1 =0.9, $-k_1'$ =-0.9

Fig. 14. Influence of weights of the original color and the transferred color. γ_1 and $-k_1'$ are shown under each result. (a) Content image and reference style image. (c)-(d) Style transfer results with varying γ_1 and $-k_1'$.

Using Eq. 11 (TE module) to repair the image, we obtain a normal new image. Although we fine-tune the weights of some structural layers and texture layers in subsequent operations, to ensure that the final result is a normal image, the weights of the three variables corresponding to Eq. 12 should be the same in the final expression. Hence, according to Eq. 11, the sum of the weights of the structural layers is 0, and the sum of the others' weights is 1. Hence, we have

$$\begin{aligned} &\gamma_1 - \gamma_2 K_1 k_1' = 0, \\ &\gamma_2 K_1 k_1 + \gamma_2 K_1' = 1, \end{aligned} \tag{26}$$

where we typically set $\gamma_1 = k_1'$, $\gamma_2 = K_1 = 1$, and $k_1 = 0.8, K_1' = 0.2$. γ_1 controls the weight of maintaining the reference style, and k_1' is the weight of subtracting the style of $\mathcal{C}$. The higher the parameters γ_1 and k_1', the more prominent the transferred reference style is in the final style transfer result, and the more completely the original style of $\mathcal{C}$ will be removed. We can adjust γ_1 and k_1' to achieve the different results shown in Fig. 14. We experimentally set $\gamma_1 = k_1' = 1$. We combine all O_j ($j \in \{R, G, G\}$) in Eq. 11 to obtain the final style transfer result O.

F. Implementation details

The computer configuration used in this paper is as follows: processor: Intel CoreTM i7-6800 k, CPU @ 3.40 GHz x 12, memory (RAM): 64.0 GB, system type: a 64-bit operating system, graphics card: GeForce GTX 1080/PCie/SSE2, and operating system: Ubuntu 18.04 LTS. The ILS filter is written in MATLAB2022B, and the rest of the code is written in Python 3.9.

This research focus on improving the video transfer outcomes rather than fast processing. Therefore, WLS has been implemented as using CPU. To compare computational time with WCT^2 which is running on GPU, we will need to first develop a GPU version of our WLS filter. For transferring images with 500 × 500 pixels, the computing time of our model is about 30 seconds. More than 95% of the calculation time is spent on WLS filtering. This time is mainly consumed by the CPU rather than GPU. In the future, we will work on the GPU version and parallel processing of WLS filtering, and try to perform real-time filtering.

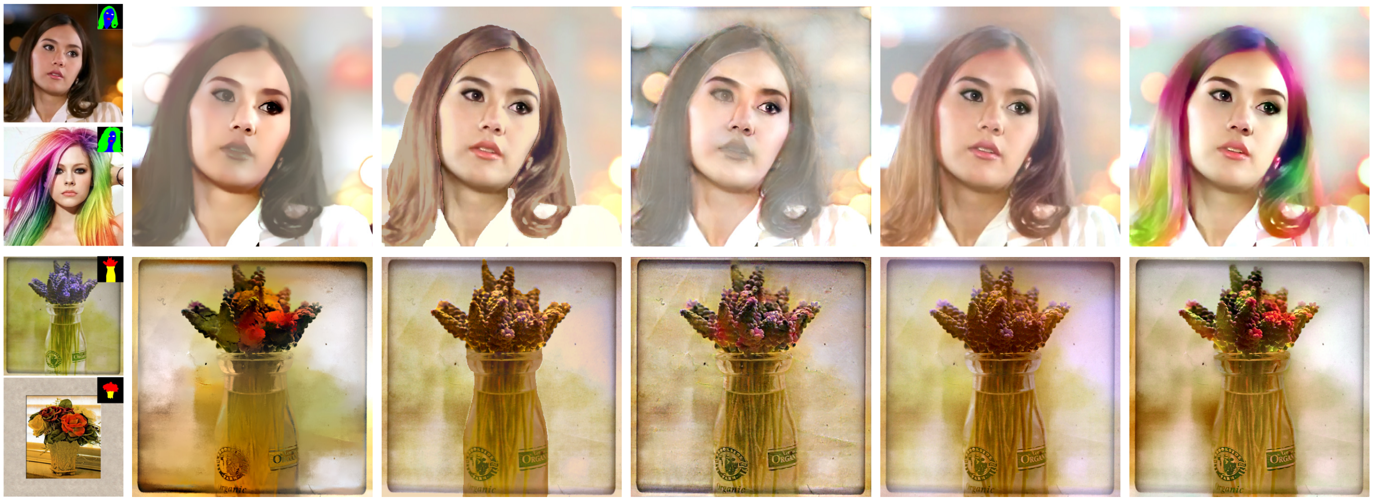
IV. Experiments

A. Comparison with state-of-the-art methods

1) Compared methods: Our approach allows for both with and without masks, (as in Fig. 15 and Fig. 16). For face images, we usually combine masks to achieve more accurate results of photorealistic face image style transfer. We select four state-of-the-art photorealistic image style transfer methods: Luan et al. [6], Yoo et al. [8], Li et al. [9], and Ding et al. [12] for style transfer comparison of face and scene-level images with masks. Fig. 15 shows the results. We also compare our method with seven state-of-the-art photorealistic image style transfer methods: Yoo et al. [8], Li et al. [9], An et al. [10], Hong et al. [11], An et al. [19], Ding et al. [12], and Wen et al. [15], for photorealistic image style transfer without masks. For the method of Hong et al. [11], we produce the photorealistic image style transfer results by running their code [1].

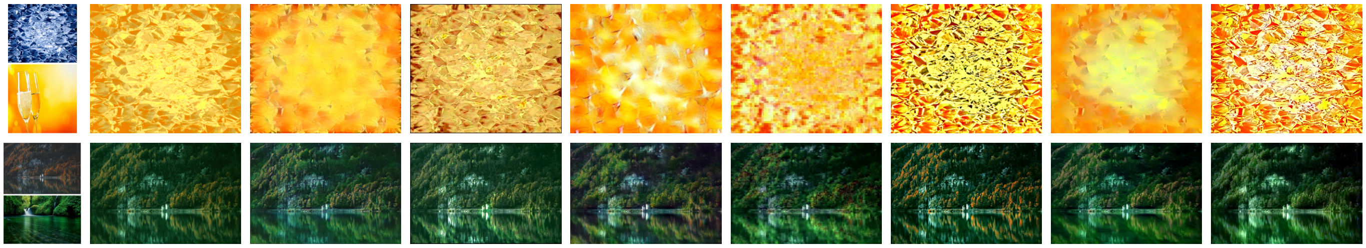
2) Visual comparison: In Fig. 15 and Fig. 16, the method of Yoo et al. [8] shows limitations in effectively handling object boundaries of the images. Results of [6], [9], [10], [11], [19], and [15] have visible distortion in some local areas. [9] does not perform style transfer well for face image. The approach described in [11] fails to yield satisfactory results for photorealistic images, as it primarily emphasizes artistic effect in the image style transfer work. Because the texture layer extracted by the AWLS filter has some original color of

[1] https://github.com/Kibeom-Hong/Domain-Aware-Style-Transfer



(a) $\mathcal{C}$,$\mathcal{S}$ (b) Luan et al. [6] (c) Yoo et al. [8] (d) Li et al. [9] (e) Ding et al. [12] (f) **Ours**

Fig. 15. Comparison with masks. (a) $\mathcal{C}$ (top), $\mathcal{S}$ (bottom), and their masks (on the upper right corner). (b)-(f) Style transfer results of Luan et al. [6], Yoo et al. [8], Li et al. [9], Ding et al. [12], and ours, respectively.



(a) (c)Yoo [8] (d) Li [9] (e) An [10] (f) Hong [11] (g) An [19] (h) Ding [12] (b) Wen [15] (i) **Ours**

Fig. 16. Comparison results without masks. (a) $\mathcal{C}$ (top) and $\mathcal{S}$ (bottom). (b)-(i) Style transfer results of Yoo et al. [8], Li et al. [9], An et al. [10], Hong et al. [11], An et al. [19], Ding et al. [12], Wen et al. [15], and ours, respectively.

$\mathcal{C}$, although method of [12] has robust texture retention ability, the results of the method of [12] show incompletely style transfer effect. On the contrary, our method produces improved photorealistic image style transfer results, which are closer to the reference colors, and have good texture preservation effect.

3) Quantitative evaluation: We use the natural image quality evaluator (NIQE) and average gradient (AG) to quantitatively evaluate the results. NIQE is a blind image quality analyzer. This evaluation depends on quantifiable deviations from statistical laws found in natural images. It does not train on human-rated distorted images, and is not exposed to distorted images. AG or ”acutance gradient” is related to the sharpness of an image, including the difference of fine details and the change of texture, as well as the overall sharpness of the image. Lower NIQE and higher AG values indicate better results. We report the results in Table I and Table II corresponding to Fig. 15 and Fig. 16, respectively. We also use SSIM and PSNR as quantitative indicators of comparative analysis. Higher SSIM and PSNR values indicate better results. In addition, we measure the style similarity between stylized and style images using Gram distance (GD) for the test dataset at the code published by Luan [6]. We report the results in Table III, Table V and Table IV, respectively. In Table I, our AG values are higher than those of other methods. Results of Li et al. [9] have the lowest NIQE values. However, our method surpasses [9] in terms of the significance of style transfer and texture preservation. In Table III, for the face images, our SSIM and PSNR values are higher than those of other methods. For the flower images, our SSIM value is lower than that of Ding et al. [12], and our PSNR value is lower than that of Li et al. [9]. However, our method surpasses their methods in terms of the significance of style transfer. In Table II, for the glass image, AG value of Li et al. [9] and NIQE value of An et al. [10] are better than ours. However, the style transfer results of [9] and [10] have noticeable texture loss. For the mountain image, NIQE value of An et al. [10] is best. However, our result has more significance of style transfer. In Table V, for the glass image, our SSIM and PSNR values are the best. For the mountain image, SSIM value of Wen et al. [15] is the best, and PSNR value of Li et al. [9] is better than ours. However, our result has more significance of style transfer.

B. User study

These four quantitative comparison methods may not be suitable for evaluating image style transfer approaches because they do not consider the significance of image style transfer. When the style transfer effect is better, the style transfer results may cause more substantial changes to the original

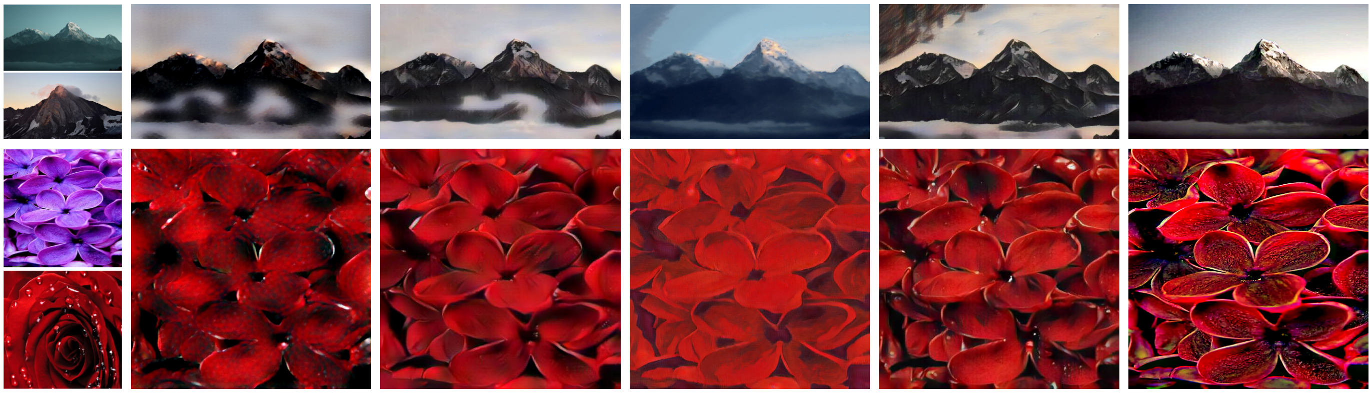

(a) $\mathcal{C}$,$\mathcal{S}$ (b) Yao [3] (c) Yu [41] (d) Svoboda [42] (e) Deng [43] (f) **Ours**

Fig. 17. Comparison results with artistic style methods. (a) $\mathcal{C}$ (top) and $\mathcal{S}$, (bottom). (b)-(f) Style transfer results of Yao et al. [3], Yu et al. [41], Svoboda et al. [42], Deng et al. [43], and ours.

TABLE I
COMPARISONS ON NIQE AND AG METRICS. THESE EVALUATIONS CORRESPOND TO THE FIRST ROW AND THE SECOND ROW IN FIG. 15, RESPECTIVELY.

	[6]	[8]	[9]	[12]	Ours
NIQE	7.50	5.75	**5.58**	5.70	5.69
AG	1.54	2.32	2.14	2.00	**2.43**
NIQE	3.34	3. 51	**2.68**	3.38	3.10
AG	3.50	4. 01	3.50	4.18	**4.45**

TABLE II
COMPARISONS ON NIQE AND AG METRICS. THESE EVALUATIONS CORRESPOND TO THE FIRST ROW AND THE SECOND ROW IN FIG. 16, RESPECTIVELY.

	[8]	[9]	[10]	[11]	[19]	[12]	[15]	Ours
NIQE	8.20	**5.39**	5.94	7.13	7.09	6.51	7.94	5.36
AG	1.62	1.81	**4.06**	1.81	3.37	3.87	1.52	3.85
NIQE	5.30	4.99	**3.16**	4.97	6.76	4.45	4.52	4.18
AG	3.26	4.46	6.71	4.29	3.02	6.70	3.35	**6.72**

TABLE III
COMPARISONS ON SSIM AND PSNR METRICS. THESE EVALUATIONS CORRESPOND TO FIG. 15.

	[6]	[8]	[9]	[12]	Ours
SSIM	0.57	0.56	0.55	0.62	**0.63**
PSNR	55.29	55.25	55.10	55.42	**55.43**
GD	**0.91**	1.12	0.93	0.95	0.96
SSIM	0.77	0.89	0.88	**0.94**	0.91
PSNR	64.38	65.87	**66.34**	65.11	66.08
GD	0.86	**0.79**	0.81	1.02	0.83

TABLE IV
COMPARISONS ON SSIM, PSNS AND GD METRICS. THESE EVALUATIONS CORRESPOND TO THE DATASET [6].

	[6]	[8]	[9]	[12]	Ours
SSIM	0.63	0.73	0.71	0.75	**0.76**
PSNR	57.81	58.95	57.72	59.93	**61.59**
GD	0.82	0.86	0.85	0.81	**0.76**

content image. The chances sometimes lead to the fact that the evaluations are not fair. Consequently, we have introduced a user study to provide a more comprehensive comparison of various style transfer methods.

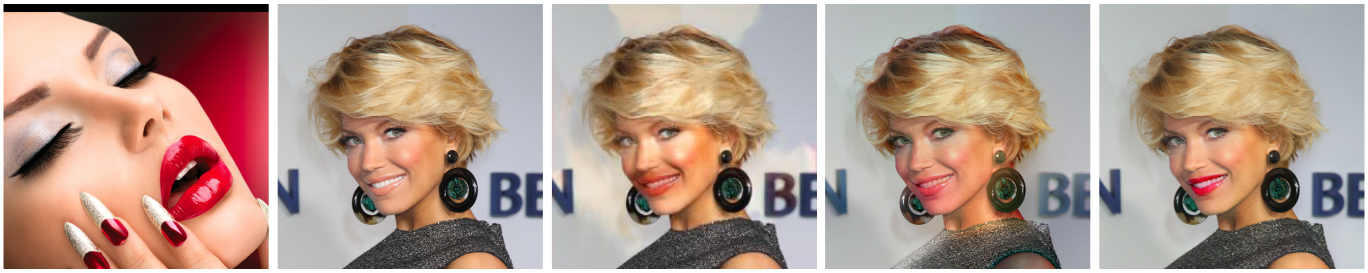
We conducted a user study involving 80 volunteers selected randomly to verify the effectiveness of these proposed methods. For each volunteer, we presented 50 sets of style transfer outcomes when using both our approach and other compared approaches, including those cited as follows: [6], [9], [8], [10], [11], [19], [12], and [15]. We carry out a survey to gather feedback on the following inquiries. (i) Clarity of details and distinct contrast. (ii) Natural and vivid color. (iii) Preservation of textures, and (iv) maintenance of photorealism. For fairness, the results generated by the eight methods are labelled as A_1, A_2, A_3, A_4, A_5, A_6, A_7 and A_8, respectively, while our result is labelled A_9. The results are displayed in Table VI.

We express the total number of votes of $A_{i'}$ on the j'-th question as $V_{i'j'}$, and evaluate each approach for an independent question in the following way. We calculate the percentage of votes PoV as $PoV = \left(\frac{V_{i'j'}}{8000}\right) \times 100\%$, where $V_{i'j'}$ can be achieved by $A_{i'}$ on the j'-th question. To provide an overall evaluation of different methods, we further calculate the percentage of votes obtained by $A_{i'}$ on $\overline{PoV} = (\sum_{j'=1}^{4} V_{i'j'})/32000 * 100\%$. In Table VI, we give the percentage of votes obtained by different methods, where Qu.x$'$ denotes the x'-th question. Table VI shows that our method achieves the best results. These inquiries indicate that human subjects prefer our results.

C. Comparison with other methods

1) Artistic style comparison: We compare our model with four recent state-of-the-art artistic style transfer methods including [3], [41], [42] and [43]. Results are presented in Fig. 17, which shows that the four competitive methods are more suitable for artistic style transfer than photorealistic style transfer.

2) comparison with makeup transfer methods: Makeup transfer methods [27], [30] do not work well for image pairs ($\mathcal{C}$ and R) without dense correspondence. The outputs of Zhang et al. [28] heavily depend on the training dataset. When the content face image is very different from the reference style



(a) $\mathcal{S}$ (b) $\mathcal{C}$ (c) Zhang et al. [28] (d) Ding et al. [12] (e) Ours

Fig. 18. Comparison of local makeup. We transfer the mouth color in $\mathcal{S}$ to $\mathcal{C}$ for visual comparison.

TABLE V
COMPARISONS ON SSIM AND PSNR METRICS. THESE EVALUATIONS CORRESPOND TO FIG. 16, RESPECTIVELY.

	[8]	[9]	[10]	[11]	[19]	[12]	[15]	Ours
SSIM	0.11	0.12	0.09	0.06	0.07	0.13	0.08	**0.14**
PSNR	51.60	52.03	51.90	51.36	51.26	51.81	51.29	**51.91**
GD	0.20	0.21	0.14	0.15	0.76	0.82	0.13	**0.09**
SSIM	0.80	0.75	0.70	0.72	0.61	0.78	**0.82**	0.80
PSNR	69.22	**69.85**	69.26	69.24	67.91	68.98	69.23	69.31
GD	0.99	0.95	0.94	0.95	0.96	0.99	0.96	**0.93**

TABLE VI
VOTING RESULTS FOR OUR APPROACH AND THE COMPARISON APPROACHES.

Methods	Qu.1	Qu.2	Qu.3	Qu.4	Overall
Luan [6]	5.43%	5.78 %	3.68%	4.78%	4.91%
Yoo [8]	6.58%	5.78 %	6.08%	6.43%	6.21%
Li [9]	9.03%	8.90 %	10.05%	9.03%	9.25%
An [10]	7.03%	7.48 %	7.35%	6.83%	7.17%
Hong [11]	4.38%	5.73%	4.18%	4.73%	4.75%
An [19]	12.03%	12.30 %	13.13%	12.95%	12.60%
Ding [12]	12.98%	12.83%	18.38%	15.05%	14.81%
Wen [15]	5.90%	3.40 %	4.30%	2.73%	4.08%
Ours	**36.68**%	**37.83**%	**32.88**%	**37.53**%	**36.23**%

Note: Qu. denotes question.

image in posture, the results are not convincing. The result of Ding et al. [12] has color overflow near the mouth, and our result has the best visual effect. Some results are shown in Fig. 18.

3) Comparison with sharpening post-processing: Our model has a good texture preservation ability. When we sharpen the style transfer results of state-of-the-art methods on iPhone, our result is still clearer than theirs. Fig. 19 shows the comparison results.

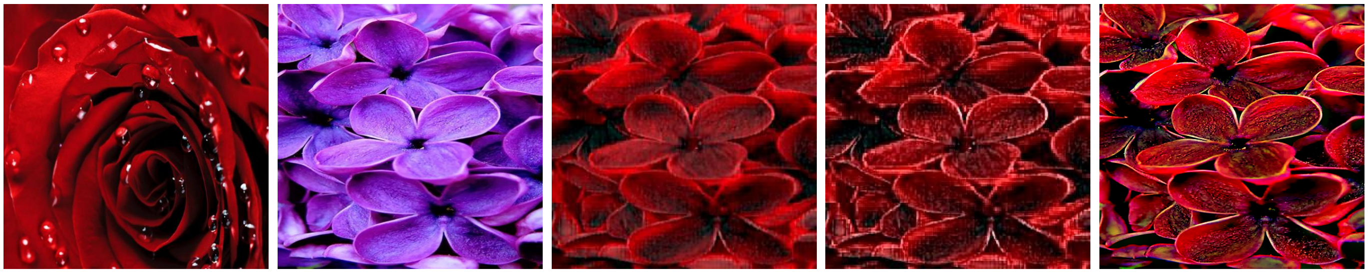
D. Discussion

1) Comparison with the LAB channel: Similar to method of [12], our method also chooses RGB channel separated technology instead of LAB channel separated technology. For our photorealistic image style transfer task, we expect to get more powerful and delicate controlling ability. Compared to LAB or HSV, RGB has three basic components to jointly determine color type, while other color spaces would use only one or two components to determine color type. In a more explicit manner of using three network branches to respectively establish R, G and B correspondence between $\mathcal{S}$ and $\mathcal{C}$ would increase the correctness and the delicacy of style transfer result. We put the correctness of style transfer as the first importance. With regard to other factors like color purity and color brightness, we believe the network could deal with them in an implicit way.

Consequently, we opt to utilize the R, G, and B channels instead of the L, A, and B channels. In order to visually compare the results, please refer to Fig. 20. Our approach can generate more natural style transfer results in RGB color space than it does in LAB color space.

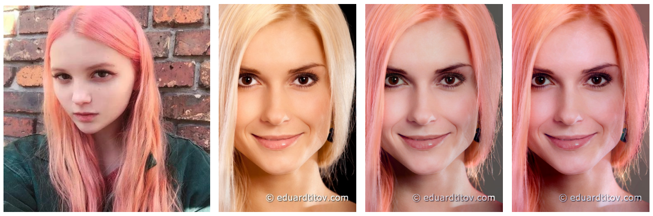
2) Comparisons with Ding et al. *[12]:*
A. Channel separation for the whole images (channel separation only at the structural layers in [12]). In this paper, we perform color transfer in the structural layer from ILS filter channel-by channel. We also compensate for the texture loss caused by TE module channel-by-channel in the textural layer.
B. Feature compensation (no loss compensation in [12]). The method used AWLS filter to smooth the input image. In the AWLS structure layer, [12] still used the AWLS filter to enhance the color transfer results of WCT^2. This operation does not result in texture loss. In this paper, we use the ILS filter to smooth the input image, and then use the AWLS filter to repair the boundary of WCT2 results. The texture layer obtained by the AWLS filter contains less texture details than that obtained by the ILS filter. When we use the AWLS texture layer to enhance the ILS texture layer, there will be less texture after repairing.
C. The position of AWLS and ILS (only one filter used in [12]). In this paper, the position of the AWLS filter and the ILS filter cannot be interchanged. The function of the ILS filter is different from that of the AWLS filter. For example, if $\mathcal{C}$ is filtered by the AWLS filter at the beginning, some original style of the input image will be remained in the texture layer at the beginning, resulting in incomplete image style transfer. AWLS filter in this paper is mainly used to repair the boundary problem of WCT^2, while the ILS filter does not have the function of image enhancement.

3) Limitations: Our method also has two limitations. First, the texture preservation of our model depends on the texture layers extracted by both the WLS filter and the ILS filter. When an image is too dark, the values of brightness and texture of the image tend to the extreme value, which makes the gradient solution ineffective. The invalid gradient solution results in the lack of useful information in texture layer, and it leads



(a) $\mathcal{S}$ (b) $\mathcal{C}$ (c) An et al. [19] (d) [19] + $\mathcal{S}h$ (e) Ours

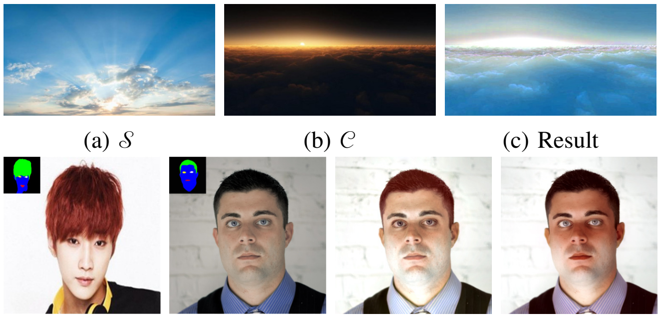
Fig. 19. Comparison with sharpening post-processing. $\mathcal{S}h$ indicates the use of the sharpening post-processing on iPhone 12.



(a) $\mathcal{S}$ (b) $\mathcal{C}$ (c) In RGB (d) In LAB

Fig. 20. Comparison of photorealistic image style transfer in different color spaces. Our method with RGB color space is able to produce more photorealistic image style transfer result with less visual artifacts than does it with LAB color space. We can see that the facial region of the result in (d) seems to be polluted by the style from the hair in (a).

to the lack of clarity in our result, as shown in the first row of Fig. 21. Second, in the absence of masks in style transfer, the selection of style transfer regions becomes arbitrary. If we conduct style transfer for face images without masks, our method fails to produce satisfactory results with noticeable artifacts. We show the two failure cases, as presented in the second row in Fig. 21.



(a) $\mathcal{S}$ (b) $\mathcal{C}$ (c) Result

(d) $\mathcal{C}$, $\mathrm{M}_{\mathcal{C}}$ (e) $\mathcal{S}$, $\mathrm{M}_{\mathcal{S}}$ (f) With M (f) Without M

Fig. 21. Instances of failure. The first row: scene-level images; the second row: face images. M denotes the masks of $\mathcal{C}$ and $\mathcal{S}$.

V. Conclusion and Future Work

This paper has introduced a novel framework towards high-quality photorealistic image style transfer. We introduced an adaptive image smoothing method for both content and reference style images, RGB channel separation module, texture enhancing module, estimation and compensation of the structural layer texture loss module, and image merging module. By combining the ILS filter and the AWLS filter, the proposed method has proved its excellent ability in extracting complex image textures and reducing color overflow in photorealistic image style transfer results. We have evaluated various scene-level images and face images to demonstrate the superior performance of our approach compared with the most advanced approaches. In the future, we plan to expand our approach to photorealistic video style transfer. Furthermore, in order to enhance the stability and vividness of the style transfer results, we intend to add illumination editing methods [44]–[46] to our approach.

Acknowledgments

We would like to thank Jiahao Xia for his invaluable help in building the dataset.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[2] C. Li and Wand, “Combining markov random fields and convolutional neural networks for image synthesis,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[3] Y. Yao, J. Ren, X. Xie, W. Liu, Y. Liu, and J. Wang, “Attention-aware multi-stroke style transfer,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 1467–1475.

[4] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, “Universal style transfer via feature transforms,” *arXiv preprint arXiv:1705.08086*, 2017.

[5] Q. Wang, S. Li, Z. Wang, X. Zhang, and G. Feng, “Multi-source style transfer via style disentanglement network,” *IEEE Transactions on Multimedia*, vol. 26, pp. 1373–1383, 2024.

[6] F. Luan, S. Paris, E. Shechtman, and K. Bala, “Deep photo style transfer,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6997–7005.

[7] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 1501–1510.

[8] J. Yoo, Y. Uh, S. Chun, B. Kang, and J.-W. Ha, “Photorealistic style transfer via wavelet transforms,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9036–9045.

[9] X. Li, S. Liu, J. Kautz, and M.-H. Yang, “Learning linear transformations for fast image and video style transfer,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 3809–3817.

[10] J. An, H. Xiong, J. Huan, and J. Luo, “Ultrafast photorealistic style transfer via neural architecture search,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 10 443–10 450.

[11] K. Hong, S. Jeon, H. Yang, J. Fu, and H. Byun, “Domain-aware universal style transfer,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 609–14 617.

[12] H. Ding, F. Luo, C. Jiang, G. Fu, Z. Chen, S. Hu, and C. Xiao, “Photo-realistic style transfer via adaptive filtering and channel seperation,” in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2627–2635.

[13] Q. Zhang, Y. Nie, L. Zhu, W. S. Zheng, and W. Zheng, "A blind color separation model for faithful palette-based image recoloring," *IEEE Transactions on Multimedia*, vol. 24, no. 2022, pp. 1545–1557, 2022.

[14] H. Mun, G.-J. Yoon, J. Song, and S. M. Yoon, "Texture preserving photo style transfer network," *IEEE Transactions on Multimedia*, vol. 24, pp. 3823–3834, 2022.

[15] L. Wen, C. Gao, and C. Zou, "Cap-vstnet: Content affinity preserved versatile style transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 300–18 309.

[16] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3985–3993.

[17] J. Y. Yijun Li, Chen Fang, "Diversified texture synthesis with feed-forward networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2732–2738.

[18] Y. Li, M. Y. Liu, X. Li, M. H. Yang, and J. Kautz, "A closed-form solution to photorealistic image stylizations," in *Proceedings of the European conference on computer vision*, 2018, pp. 1–16.

[19] J. An, S. Huang, Y. Song, D. Dou, W. Liu, and J. Luo, "Artflow: Unbiased image style transfer via reversible neural flows," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021.

[20] W. Liu, P. Zhang, X. Huang, J. Yang, C. Shen, and I. Reid, "Real-time image smoothing via iterative least squares," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 3, pp. 1–24, 2020.

[21] R. Rothe, R. Timofte, and L. V. Gool, "Dex: Deep expectation of apparent age from a single image," in *ICCVW*, 2016, pp. 252–257.

[22] M. M. Tsung-Yi Lin and S. Belongie, "Microsoft coco: Common objects in context," in *Proceedings of the European conference on computer vision*, 2014, p. 740755.

[23] J. Xia, M. Xu, H. Zhang, J. Zhang, W. Huang, H. Cao, and S. Wen, "Robust face alignment via inherent relation learning and uncertainty estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[24] H.-H. Zhao, P. L. Rosin, Y.-K. Lai, and Y.-N. Wang, "Automatic semantic style transfer using deep convolutional neural networks and soft masks," *The Visual Computer*, vol. 36, no. 7, pp. 1307–1324, 2020.

[25] T. Chen, W. Xiong, H. Zheng, and J. Luo, "Image sentiment transfer," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 4407–4415.

[26] T. Xiao, J. Hong, and J. Ma, "Elegant: Exchanging latent encodings with gan for transferring multiple face attributes," in *Proceedings of the European conference on computer vision*, 2018, pp. 168–184.

[27] S. Gu, J. Bao, and Yang, "Mask-guided portrait editing with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 3436–3445.

[28] H. Zhang and Chen, "Disentangled makeup transfer with generative adversarial network," *arXiv preprint arXiv:1907.01144*, 2019.

[29] S. Liu, X. Ou, and Qian, "Makeup like a superstar: Deep localized makeup transfer network," *arXiv preprint arXiv:1604.07102*, 2016.

[30] T. Li, R. Qian, and Dong, "Beautygan: Instance-level facial makeup transfer with deep generative adversarial network," in *Proceedings of the 26th ACM international conference on multimedia*, 2018, pp. 645–653.

[31] L. Dai, M. Yuan, F. Zhang, and X. Zhang, "Fully connected guided image filtering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015, pp. 352–360.

[32] X. Tan, C. Sun, and T. D. Pham, "Multipoint filtering with local polynomial approximation and range guidance," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2941–2948.

[33] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, pp. 1–10, 2008.

[34] J. T. Barron and B. Poole, "The fast bilateral solver," in *Proceedings of the European conference on computer vision*, 2016, pp. 617–632.

[35] Q. Fan, D. Chen, L. Yuan, G. Hua, N. Yu, and B. Chen, "A general decoupled learning framework for parameterized image operators," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[36] Q. Fan, J. Yang, D. Wipf, B. Chen, and X. Tong, "Image smoothing via unsupervised learning," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–14, 2018.

[37] J. Y. Yim, Jonghwa, "Filter style transfer between photos," in *Proceedings of the European conference on computer vision*, 2020, pp. 103–119.

[38] S. Bi, X. Han, and Y. Yu, "An l1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition," in *International Conference on Computer Graphics and Interactive Techniques*, 2015.

[39] Y. HaCohen and E. Shechtman, "Non-rigid dense correspondence with applications for image enhancement," *ACM Transactions on Graphics*, vol. 30, no. 4, pp. 1–10, 2011.

[40] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer Graphics and Applications*, vol. 21, no. 5, pp. 34–41, 2002.

[41] Y. W. Yulun Zhang, Chen Fang, "Multimodal style transfer via graph cuts," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5943–5951.

[42] J. Svoboda, A. Anoosheh, C. Osendorfer, and J. Masci, "Two-stage peer-regularized feature recombination for arbitrary image style transfer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020, pp. 2761–2776.

[43] Y. Deng, F. Tang, W. Dong, W. Sun, F. Huang, and C. Xu, "Arbitrary style transfer via multi-adaptation network," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 2719–2727.

[44] Z. Bao, C. Long, G. Fu, Y. Li, J. Wu, D. Liu, and C. Xiao, "Deep image-based illumination harmonization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022, pp. 18 542–18 551.

[45] Z. Chen, C. Long, L. Zhang, and C. Xiao, "Canet: A context-aware network for shadow removal," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4743–4752.

[46] G. Fu, Q. Zhang, L. Zhu, P. Li, and C. Xiao, "A multi-task network for joint specular highlight detection and removal," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021, pp. 7752–7761.