Structure-Preserving Neural Style Transfer

Ming-Ming Cheng*, Xiao-Chang Liu*, Jie Wang, Shao-Ping Lu, Yu-Kun Lai and Paul L. Rosin

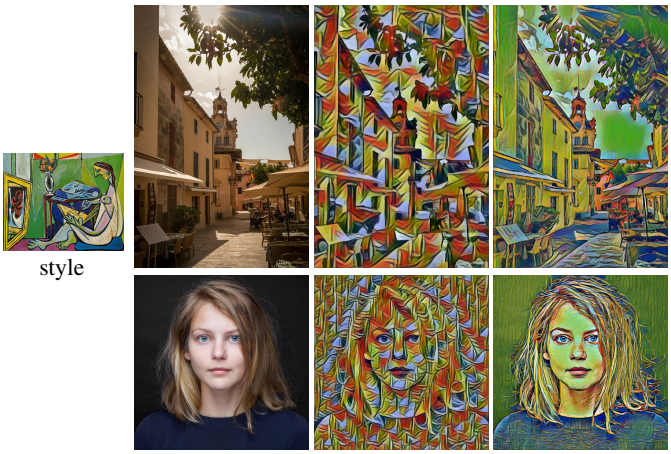
***Abstract*—State-of-the-art neural style transfer methods have demonstrated amazing results by training feed-forward convolutional neural networks or using an iterative optimization strategy. The image representation used in these methods, which contains two components: style representation and content representation, is typically based on high-level features extracted from pre-trained classification networks. Because the classification networks are originally designed for object recognition, the extracted features often focus on the central object and neglect other details. As a result, the style textures tend to scatter over the stylized outputs and disrupt the content structures. To address this issue, we present a novel image stylization method that involves an additional structure representation. Our structure representation, which considers two factors: i) the global structure represented by the depth map and ii) the local structure details represented by the image edges, effectively reflects the spatial distribution of all the components in an image as well as the structure of dominant objects respectively. Experimental results demonstrate that our method achieves an impressive visual effectiveness, which is particularly significant when processing images sensitive to structure distortion, *e.g.* images containing multiple objects potentially at different depths, or dominant objects with clear structures.**

***Index Terms*—Style transfer, structure preserving, deep learning, neural network, local structure, global structure**

I. Introduction

This paper considers the problem of stylizing images using neural networks. Broadly speaking, style transfer combines two images, making the results similar to one image in respect to style while remaining consistent with the content of the other one, *e.g.* Fig. 2. To date, many impressive results have been achieved, but it should be noted that proper image representations are important elements in generating impressive visual results.

Benefiting from their strong ability on image representation, deep neural networks quickly became a popular tool for image stylization, leading to the development of many neural style transfer methods in recent years. Gatys *et al.* [7], [9], [10] use image representations derived from intermediate activations of a pre-trained classification network, and generate a stylized image through an iterative process. Some works [20], [41], [42] avoid the slow optimization procedure by learning feed-forward networks. Other studies [17], [47] have considered how to transfer multiple styles with only one model. However, all these methods largely neglect the inherent structural information present in the scene that is viewed in the image.



(a) Content (b) Johnson *et al.* (c) Ours

Fig. 1. State-of-the-art neural style transfer methods, *e.g.* Johnson *et al.* [20], tend to scatter the textures over the stylized outputs, and ignore the structure information in an image. Our method better preserves the spatial distribution (first row) and the structure of dominant object (second row).

A problem when applying style transfer to challenging input images with complex spatial layouts is that the synthesized images tend to distribute style elements evenly across the whole image, making the holistic structure become unrecognizable. This is particularly true for images of scenes covering a wide range of depths, and the results are not entirely satisfactory (see an example in Fig. 1b). For inputs with prominent fundamental characteristics or sensitive to structure distortion, the uniformly distributed textures further obscure weak details and destroy the original structure.

These problems are mostly caused by the choice of image representations. Current methods are based on the observation that the features at intermediate layers of the pre-trained classification networks is a powerful image representation. However, such pre-trained networks were originally designed for classification, and hence the high-level features often focus on the primary target and neglect other details. Also, as pointed out in [39], the VGG architectures, used by Gatys *et al.* and others, are trained on small size images (224×224 pixels), in which features will consequently be small scale. Therefore, current representations are not sufficient for representing image details and capturing the image structures (see Fig. 3) that are necessary for good style transfer.

Some previous works have demonstrated that preserving structures can produce attractive artistic results [13], [36], [46]. Considering that a depth map can effectively capture the global structure of the scene [16], [40], the early version of this paper [30] integrates depth as an additional representation to preserve overall image layout during style transfer.

A preliminary version of this paper has been published in NPAR 2017 [30].

X.C Liu, Jie Wang, M.M. Cheng, S.P. Lu are with the TKLNDST, College of Computer Science, Nankai University, Tianjin, 300350, China. Email: cmm@nankai.edu.cn.

Y.-K. Lai and P. L. Rosin are with Cardiff University.

* These two authors contributed equally to this work.

(a) Content (b) Gatys *et al*. [10] (c) Johnson *et al*. [20]

(d) Style (e) AdaIN [17] (f) Ours

Fig. 2. Which image stylization seems best to you? Human faces are very sensitive to structure distortion, a loss of structure will disfigure them. The result in (b) is generated using a slow optimization process. Results (c) and (e) are both from fast methods. These three methods, which obtain their image representations with the help of classification networks, suffer from some similar problems: the style textures scatter over the results and disrupt the original structure, and some content details are missing. Our method use a structure-enhanced image representation to produce results shown in (f).

In this work, we go further and use a structure-enhanced image representation to control style transfer. Compared with the preliminary version [30], this work introduces an edge detection network as a local structure refinement network to coordinate work with the original global structure extraction network. Under the guidance of these two networks, we provide a trade-off between the global structure and local structure. Experiments show that when processing images, our method can yield pleasant results that effectively preserve the structures and key details. A user study also shows that our method performs well on keeping the structure consistency, and our stylization effects are preferred by the participants. So our method is very suitable for processing images which are sensitive to structure distortion (*e.g*. human faces, buildings).

To sum up, the contributions of this paper are:

- Our work demonstrates that image representations play a very important role in image transformation, and different stylization results can be generated by designing appropriate image representations;
- Moreover, we introduce a novel structure-enhanced image representation framework for style transformation, which improves the visual quality under the guidance of a global structure extraction network and a local structure refinement network.

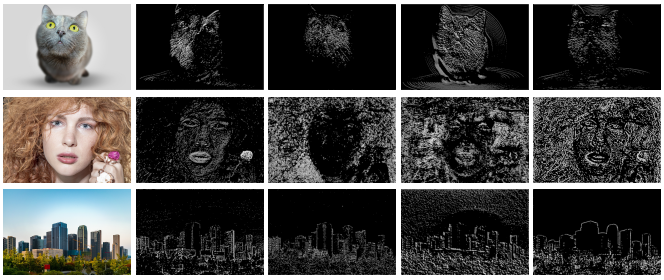
II. Related Work

As stated in previous works [10], [20], [42], two key issues in stylization are: 1) how to get an appropriate representation for the image style and content; 2) how to quantitatively measure the style/content similarity between two images. Another factor that affects the performance is the structure of image generation networks.

Deep Image Representations. For image representation, feature extraction is a crucial step. How to extract ideal features that can reflect the images as completely as possible is especially important. Traditional methods, whether parametric [14], [21], [35] or non-parametric methods [5], [6], [15], all use the image representations based on hand-crafted low-level features.

The development of Deep Convolutional Neural Networks (CNN) [23] breaks the limitation of traditional image representations. Gatys *et al*. [8]–[10] use the high-level features extracted by a trained classifier VGG [38] to represent the content of an image, and use features' Gram matrices to represent the style. After an iterative optimization process, they turned a white noise initialization into an amazing stylized image. This approach, especially the proposed image representation, is elegant and effective, and has been widely adopted by many subsequent works. However, the iterative procedure means it takes a considerable amount of time to generate a stylized image.

Some works [20], [41], [42] avoid the slow optimization procedure by learning a feed-forward network for every style. [12] improves stability in neural style transfer. Further improvements are mainly concentrated on model flexibility [1], [4], [26], [47] or processing speed [2], [17], they try to integrate multiple styles in one model, and further accelerate the processing speed at the same time. Meanwhile, photo-realistic image stylization methods [27], [32], [34] also widely use deep image representations. Among all the works, Johnson's method [20] stands out by way of its fast speed whilst achieving results with satisfactory quality. By pre-training a feed-forward network rather than directly optimizing the loss functions as in [10], Johnson's method is orders of magnitude



(a) Content (b) Activations

Fig. 3. Activations currently used in stylization for image representation: We feed images into VGG-16 and visualize some activations of the network ($relu1\_2$, $relu2\_2$, $relu3\_3$ and $relu4\_3$). Examining these activations we discover that these features do not capture the global and local structures well.

more efficient for stylizing new input images. It is worth noting that generative adversarial networks (GANs) [11] have also achieved impressive results in image generation. In GAN training, a generator will be trained to deceive a discriminator which in turn tries to distinguish between generated samples and real samples. GAN based solutions (*e.g.* [48]) perform well in collection style transfer, in which the target style is defined by a collection of images, whereas example-guided style transfer methods (*e.g.* [1], [8], [20], [42]) are suitable when the target style comes from a single example. Moreover, training a GAN is more challenging compared to classical network training which uses standard loss functions (*e.g.* log-loss or squared error that have closed forms). Therefore, in this work we focus on example-guided neural style transfer with only a single style example, which is not suitable for GAN-based methods. Finally, interested readers are referred to a comprehensive survey [19] on more recent deep learning based style transfer.

It seems that neural style transfer is becoming more and more powerful. But in fact, the quality of the results has not been significantly improved. The limitation of current image representations is one major cause of this problem. As shown in Fig. 3, they do not represent image details and structures sufficiently well. In this paper, we propose a new structure-enhanced representation to make up for that deficit, and build our structure-preserving style transfer based on [20].

Similarity Measure Methods. After obtaining the image representation, the next step is to find appropriate methods to quantitatively measure similarity. Generally, the similarity measure in style transfer is a distance with dimensions that are features of the images. A small distance is associated with a high degree of similarity and vice versa.

There are many similarity distance measures. In style transfer, the Euclidean distance is usually used to compute the content similarity, and this meets people's intuition. For style similarity, most of the works choose to compute the Frobenius norm between the Gram matrices.

Recently Li *et al*. [28] demonstrate that style transfer can be considered as a distribution alignment process from the content image to the style image, and matching the feature maps of the images can be seen as minimizing the Maximum Mean Discrepancy (MMD) with the second order polynomial kernel. The two similarity measures we just mentioned are simply special cases of minimizing the MMD. There are also works using other loss functions, such as MRF loss [24], [25], histogram loss [44], *etc*. The details of similarity measure methods used in this paper are in Sec. III-A.

Image Generation Networks. The speed of optimization-based methods is slow due to the iterative optimization procedure. In order to reduce the computation burden and expedite the process, Johnson *et al*. [20] propose to build a feed-forward network. By pre-training a network rather than directly optimizing the loss functions, they improve the efficiency of stylizing new input images by several orders of magnitude. Later works [1], [17], [42], [47] mostly adopt a similar approach.

Recently, SqueezeNet, designed by Iandola *et al*. [18] achieves AlexNet-level accuracy on ImageNet with 50 times fewer parameters. This inspires us to design a slim image generation network in a similar way, enabling the proposed framework to be efficiently applied for stylizing videos.

III. Method

As shown in Fig. 4, our system is composed of three main parts: two representation subnets ϕ_0 and ϕ_1, and a generator network f_W. The representation networks are used to define four loss functions: l_1, l_2, l_3 and l_4, where l_1 and l_2 are based on ϕ_0, and correspond to the style loss and content loss, also denoted as l_{style} and $l_{content}$ respectively. l_3 and l_4 are based on ϕ_1, and correspond to the depth loss l_{depth} and edge loss l_{edge}. The image transformation network is a deep residual convolutional neural network parametrized by weights W. It transforms an input image x into an output image $\hat{y}$ via the mapping $\hat{y} = f_W(x)$. Each loss function computes a scalar value $l_i(\hat{y}, y_i)$ measuring the difference between the output image $\hat{y}$ and a target image y_i ($i = 1, 2, 3, 4$ corresponding to content, style, depth and edge images).

The image transformation network is trained using stochastic gradient descent to minimize a weighted combination of the loss functions:

$$W^* = \arg\min_W \mathbf{E}_{x,\{y_i\}}[\sum_{i=1}^{4} \lambda_i l_i(f_W(x), y_i)] \quad (1)$$

The four loss functions fall into two categories: perceptual loss (l_{style} and $l_{content}$) and per-pixel loss (l_{depth} and l_{edge}). Perceptual loss functions, based on high-level features extracted from pre-trained networks, are used to measure high-level perceptual and semantic differences between images. Compared with per-pixel losses, perceptual losses measure image similarities more robustly. This works because according to some recent works (*e.g.* , [33], [37]), the convolutional neural networks pre-trained for image classification have already learned to encode the perceptual and semantic information. In contrast, per-pixel loss is more suitable when we have a ground-truth target that the network is expected to match. This is suitable for the depth and edge losses, as relative depth and edge maps can be estimated from the content and synthesized images. In our method, ϕ_0 is a pre-trained image classification network, and ϕ_1 is composed of a single-image

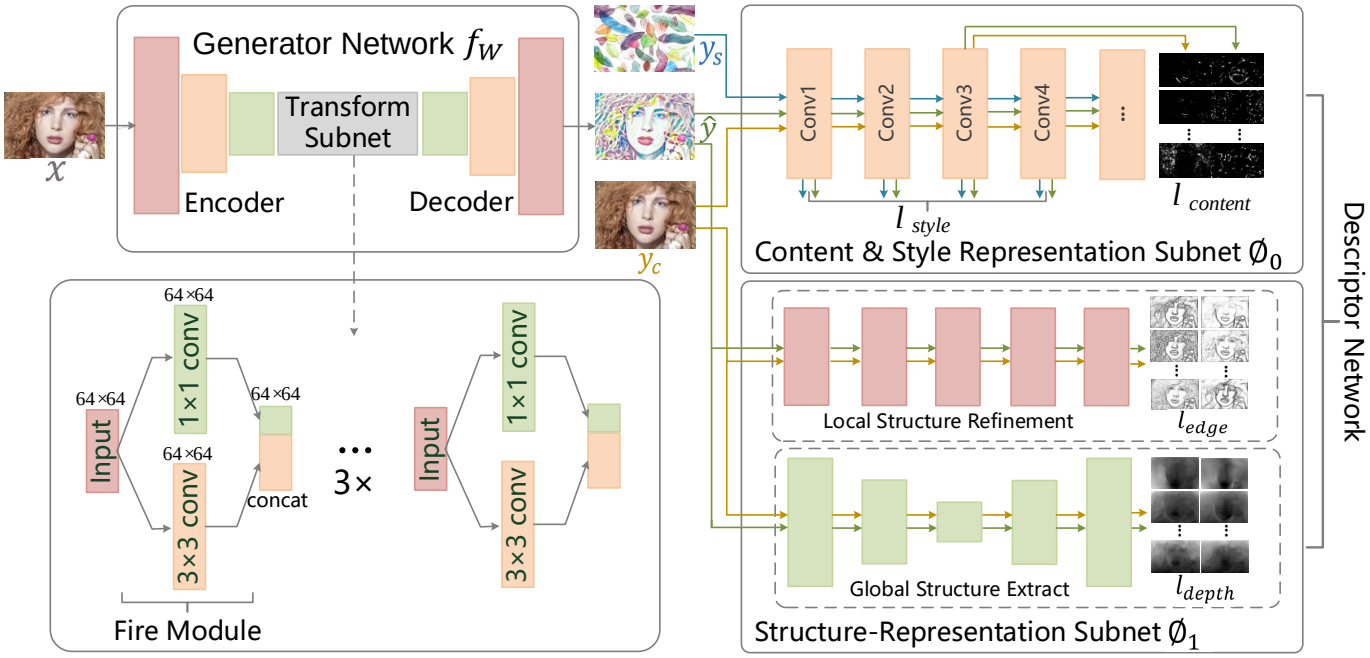


Fig. 4. Overview of our network architecture. The left side shows the generator network f_W, which transforms an input image x into $\hat{y}$ via the mapping $\hat{y} = f_W(x)$. On the right is the descriptor network. The descriptor network is used to define four loss functions: l_{style}, $l_{content}$, l_{edge} and l_{depth}, where l_{style} and $l_{content}$ are based on the Content & Style Representation Subnet ϕ_0, and correspond to the style loss and content loss. l_{edge} and l_{depth} are based on the Structure-Representation Subnet ϕ_1, and correspond to the depth loss and edge loss. y_s and y_c are the style target and content target respectively.

depth perception network [3] and a holistically-nested edge detection network [45].

In the training phase, we pass each input image x through the image transform network f_W and obtain the synthesized image $\hat{y}$. To measure the total loss, the input image x also serves as the content target y_c. The user supplied style image is treated as the style target y_s. The style reconstruction loss l_{style} is produced by comparing each $\hat{y}$ with y_s in the loss network ϕ_0, and the content reconstruction loss $l_{content}$ is produced by comparing each $\hat{y}$ with y_c in the same loss network ϕ_0. The depth reconstruction loss l_{depth} and edge reconstruction loss l_{edge} are produced by an additional depth prediction network and an edge detection network through comparing the output of $\hat{y}$ and y_c in ϕ_1, with the aim of making the stylized image retain depth and edge outputs consistent with the content.

A. Generator Network

The three parts of the system form two networks: a *Generator network* and a *Descriptor network*. We train the *Generator network* under the guidance of the global structure extraction and local structure refinement network, using a Structure-Enhanced image representation based on the correlation statistics inside the *Descriptor network*.

The generator network includes three sub-networks: *encoder subnet, transform subnet* and *decoder subnet*. We use the generator network to transform input images. Generally, each layer in the network is equivalent to a non-linear filter bank. With the increase of the layer's depth, the complexity of the filter bank increases. Hence the input image x is encoded in each layer of the network by the filter responses to that image.

Inputs and Outputs In the training phase, the input and output are both color images of size 256×256 with 3 color channels. Since the image transformation network is fully-convolutional, there is no limit to the size of test images.

Encoder and Decoder The major function of the *encoder subnet* is to map high-dimensional input images to the low dimensional space. By doing this, the latter calculation is greatly decreased. Specifically, we use one stride-1 convolution layer and two stride-2 convolution layers to down-sample the input images. In a symmetric manner, in the decoding stage, the *decoder subnet* reconstructs the original size images from the outputs of the *transform subnet*.

After these processing steps, the size of the image is preserved, but this procedure comes with two advantages: On the one hand, after down-sampling, we can use a larger network for the same computational cost. For instance, the computational cost of a 3×3 convolution with C filters on an input of size $H \times W \times C$ is equal to a 3×3 convolution with DC filters on an input of shape $\frac{H}{D} \times \frac{W}{D} \times DC$, where D is the down-sampling factor. On the other hand, down-sampling gives larger effective receptive fields with the same number of layers. For instance, without down-sampling, each additional 3×3 convolutional layer increases the effective receptive field size by 2. After down-sampling by a factor of D, the effective receptive field size increases to $2D$. In general, the larger the receptive fields, the better the style transfer results are.

Transform subnet *Transform subnet* is the core of the generator network, which shoulders the task of transforming the encoded images. Different from the existing methods, we introduce *Fire modules* [18] into the architecture to reduce

calculation and improve efficiency. By using the Fire module, Iandola *et al*. [18] achieve AlexNet-level accuracy on ImageNet with 50 times fewer parameters. We adopt a similar way to build our models so that they can be applied to videos.

A Fire module is comprised of: a *squeeze* convolution layer (which has only 1×1 filters), feeding into an *expand* layer that has a mix of 1×1 and 3×3 convolution filters. We illustrate this in Fig. 4. The main strategies of designing such architectures are:

1. Replace 3×3 filters with 1×1 filters. Using this module, on the one hand, the 3×3 filters are replaced with 1×1 filters (a 1×1 filter has 9 times fewer parameters than a 3×3 filter).

2. Decrease the number of input channels to 3×3 filters. The total quantity of parameters in a layer which is comprised entirely of 3×3 filters is: (number of input channels)$\times$(number of filters)$\times(3 \times 3)$. So in order to reduce the quantity of parameters, in addition to reducing the number of 3×3 filters, we still need to decrease the number of input channels to 3×3 filters.

By this scheme a network providing equivalent effects can be achieved, but the number of parameters is greatly decreased.

B. Descriptor Network

The descriptor network consists of two subnetworks: a Content & Style Representation Network ϕ_0 and a Structure-Representation Network ϕ_1. We use the descriptor network to obtain the image representation.

Content & Style Representation Network. Following the approach in [7], [10], [20], we use the pre-trained VGG [38] as ϕ_0 to define two loss functions $l_{content}$ and l_{style}, corresponding to the content difference and style difference.

Specifically, $l_{content}$ is defined as the (squared, normalized) Euclidean distance of activations in selected layers of ϕ_0, and l_{style} is the squared Frobenius norm of two Gram matrices.

Assume layer l of the VGG network has N_l distinct filters, and the size of each feature response is $H_l \times W_l$, where H_l and W_l are respectively the height and width of the feature map in layer l. The responses in such a layer can be represented by a matrix:

$$F_l \in \mathbb{R}^{(H_l \times W_l) \times N_l} \tag{2}$$

and each value $F^l_{(i,j),k}$ is the activation of the k^{th} filter at position (i, j) in layer l.

Then the content difference between x_1 and x_2 in the l^{th} layer is:

$$l_{content}(x_1, x_2) = \frac{1}{H_l W_l N_l} \|F_l(x_1) - F_l(x_2)\|_2^2 \tag{3}$$

The style difference in this layer is:

$$l_{style}(x_1, x_2) = \|G_l(x_1) - G_l(x_2)\|_F^2 \tag{4}$$

where G_l is the Gram matrix, an $N_l \times N_l$ symmetric matrix, and $G^l_{cc'}$ is the normalized inner product of the c^{th} and c'^{th} vectorized feature maps in layer l:

$$G^l_{c,c'}(x) = \frac{1}{H_l W_l N_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} F^l_{(h,w),c}(x) F^l_{(h,w),c'}(x) \tag{5}$$

Structure-Representation Network. The structure representation network ϕ_1 consists of two sub-networks: a global structure extraction network and a local structure refinement network. They are designed to compensate for the deficiency of the content & style representation network in capturing and holding structures.

Due to their ubiquity, edge structures are particularly appropriate to represent the local structure. We take the holistically-nested edge detection (HED) system [45] as the local structure refinement network. HED is an end-to-end edge detector, which takes an image as input and directly produces the edge map image as output. It can efficiently generate multi-level perceptual features, and shows promising results in performing image-to-image learning by combining multi-scale and multi-level visual responses. In our implementation, we use its edge responses to represent local structures. l_{edge}, which stands for the local-structure difference of two images x_1 and x_2, is calculated as the Euclidean distance of activations in a selected k^{th} layer of local structure refinement network $\mathcal{E}$:

$$l_{edge}(x_1, x_2) = \|\mathcal{E}_k(x_1) - \mathcal{E}_k(x_2)\|_2^2 \tag{6}$$

The global structure extraction network is taken from a single-image depth perception network [3], which takes an entire image as input and directly predicts pixel-wise depth. The depth map is an important characteristic of an image and is well suited for reflecting the global structure, since it contains 3D feature information about the objects. l_{depth} is calculated in the same way as l_{edge}.

The structure reconstruction loss $l_{structure}$ is the weighted combination of two parts:

$$l_{structure} = \alpha \cdot l_{depth} + \beta \cdot l_{edge} \tag{7}$$

Under the guidance of the global structure extraction network and the local structure refinement network, the local and global structures are effectively captured, and they are clearly reflected in the final results (See Figs. 5 and 6). Some more details about the structure representation network are provided in Sec. V-D.

IV. Network Learning

Given that ground-truth is generally unavailable for style transfer, we choose to minimize differences judged by the descriptor network.

As previously stated, the content difference $l_{content}$ and style difference l_{style} are described by a pre-trained classification network ϕ_0, and the structure reconstruction loss $l_{structure}$ is based on the Structure-Representation network ϕ_1.

Let the generator network be denoted by f parametrized by weights W, it transforms an input image x_c into an output image $\hat{y}$ via the mapping $\hat{y} = f_W(x_c)$. Network learning adjusts the parameters W through minimizing a weighted sum loss using stochastic gradient descent:

$$\begin{aligned} W^* = \arg\min_W \mathbf{E}_{y_c, y_s} [&\lambda_{content} l_{content}(\hat{y}, y_c) \\ &+ \lambda_{style} l_{style}(\hat{y}, y_s) \\ &+ \lambda_{structure} l_{structure}(\hat{y}, y_c) \\ &+ \lambda_{TV} l_{TV}] \end{aligned} \tag{8}$$

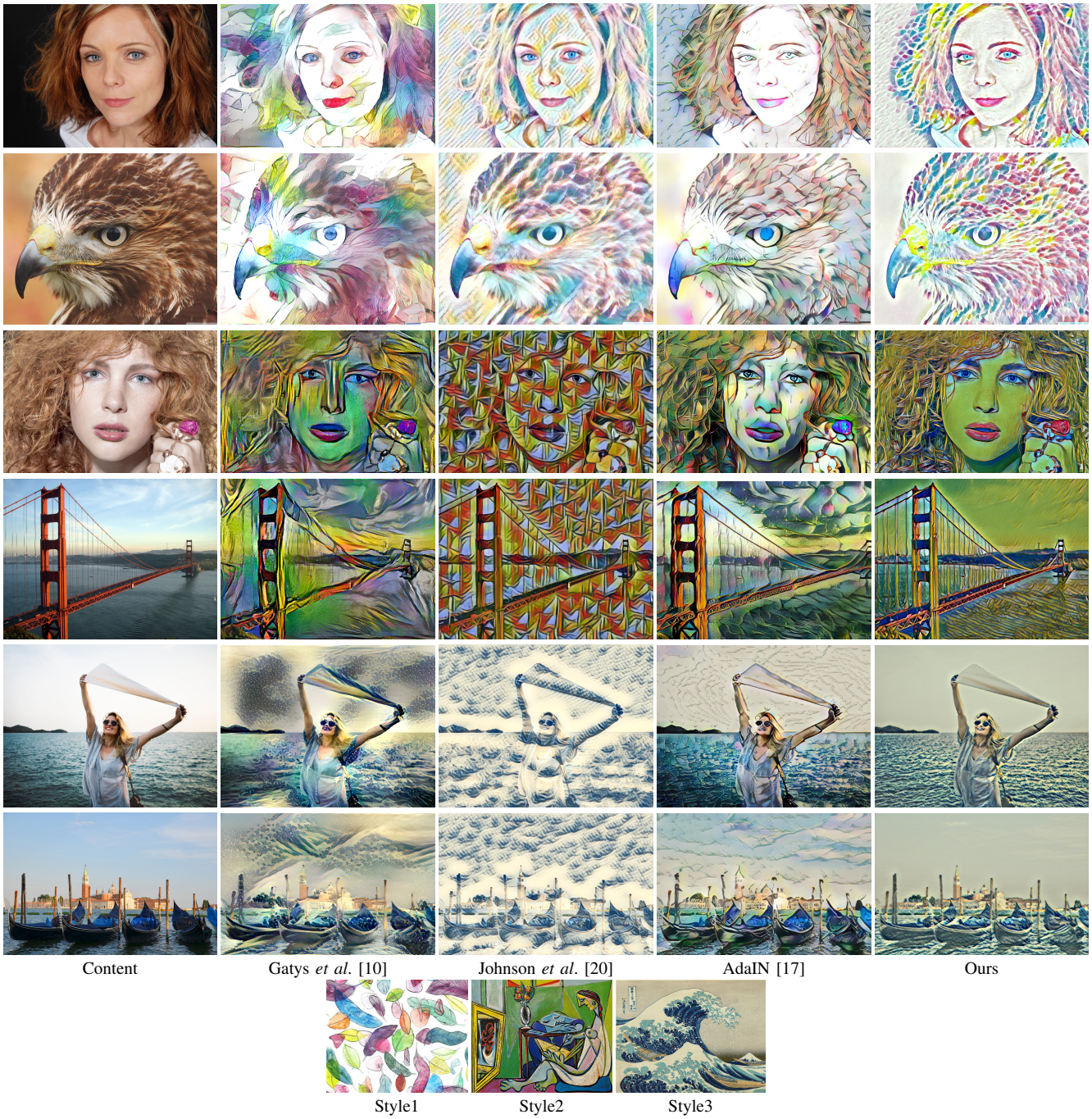


Fig. 5. Comparison with other style transfer methods. Compared with the results of other methods, our results preserve the details and structures well. The effects are more pronounced on human faces since they are very sensitive to structure distortion.

where y_c and y_s are content target and style target respectively. l_{TV} is the total variation regularization, used in previous works [1], [20], [47] to encourage the smoothness of the generated images.

V. Experiments

In this section, we provide the training details and compare our method with other CNN-based style transfer approaches [10], [17], [20], [24], [32], [34], [42]. For the sake of fairness, some results are taken directly from their papers.

A. Training Details

We choose Microsoft COCO [29] as the training dataset. The activations at layer $relu3\_2$ of VGG-16 network are used to compute $l_{content}$, and layers $relu1\_2$, $relu2\_2$, $relu3\_3$ and $relu4\_3$ are used to compute l_{style}. The structure reconstruction loss $l_{structure}$ is computed at the output layer of the structure representation network. We use Adam [22] for optimization with a learning rate of 1×10^{-3}. The default parameters of α and β in Eq. 7 are both 5, and $\lambda_{content}$, λ_{style}, $\lambda_{structure}$ and λ_{TV} in Eq. 8 are 1, 5×10^{-2}, 1×10^{-2} and 1×10^{-3} respectively. As illustrated in Fig. 4,

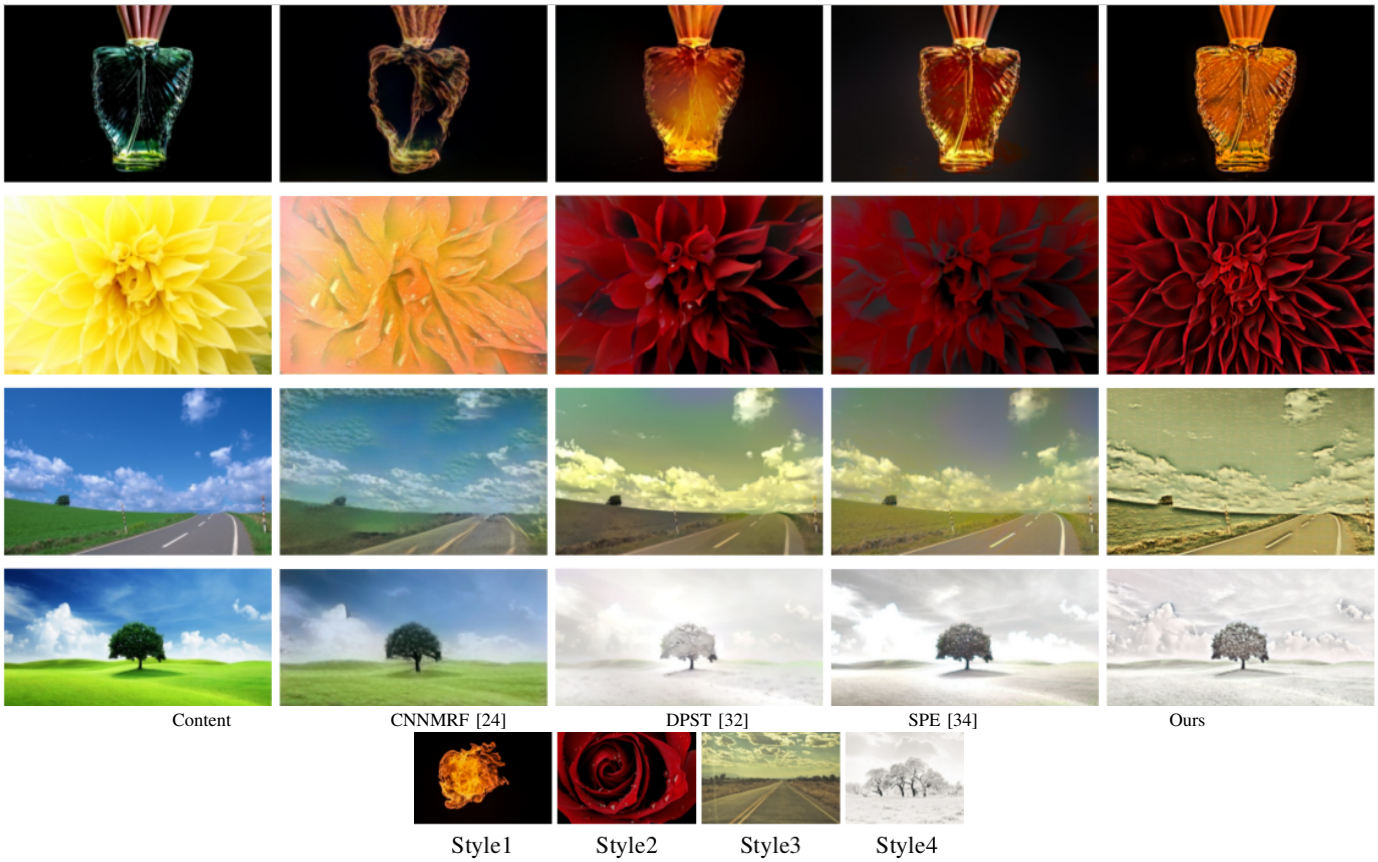


Fig. 6. Comparison with photorealistic style transfer methods. Here [32], [34] are designed for the special cases that the content and the style image share highly similar layout and semantic components. Our method can also work well in such cases.

both our global and local networks are trained jointly. The contents and outputs of the generator network are fed into the structure-representation subnet. During the training procedure, the Content & Style Representation subnet and the Structure-Representation subnet are kept fixed, and the parameters of the generator network are updated. Thus, due to the interactions between the global and local sub-networks, after the optimization, the parameters of the generator network are kept fixed, and it acquires the ability to maintain structure consistency whilst obtaining artistic effects.

B. Comparisons and Analysis

Comparison on Images. We show some comparison results in Figs. 5, 6 and 7. From the perspective of results, all the approaches are quite distinct from each other, and a strong sense of structure would be the viewers' first impression of our results. The characteristic of retaining key details has also been reflected.

First, our approach is capable of providing high contrast between the foreground and background. This is due to the global structure extraction network which provides strong global structural information, which facilitates the ability of distinguishing objects at different depths. In particular, this property makes our approach more suitable for close-up shots, such as the face and the bird (See Fig. 5). The other methods neglect the original structures of the content images, yielding results with uniformly distributed styles that deemphasize the structures in the scenes. This makes them unsuitable for processing images which contain rich spatial information.

Second, even within foreground/background regions, our approach stylizes different areas differently, providing a sense of overall balance and harmony. For example, we introduce almost no textures to the model's face (See Figs. 2 and 5), and only make some small changes on the main features (eyes, mouth, *etc*), but put great emphasis on the hair. This is because the local edge details of these areas are different from each other (rich in the hair and less in the face), and local-

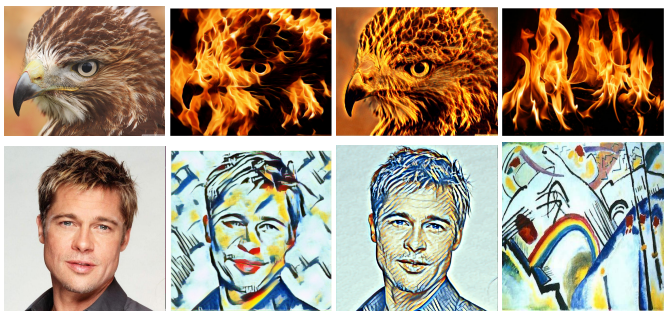


Fig. 7. Comparison with Ulyanov *et al*. [42], which introduces an instance-normalization layer to improve the performance of the deep network generators and capture the abstract feelings. Our method shows less abstraction due to the emphasis on preserving structures.

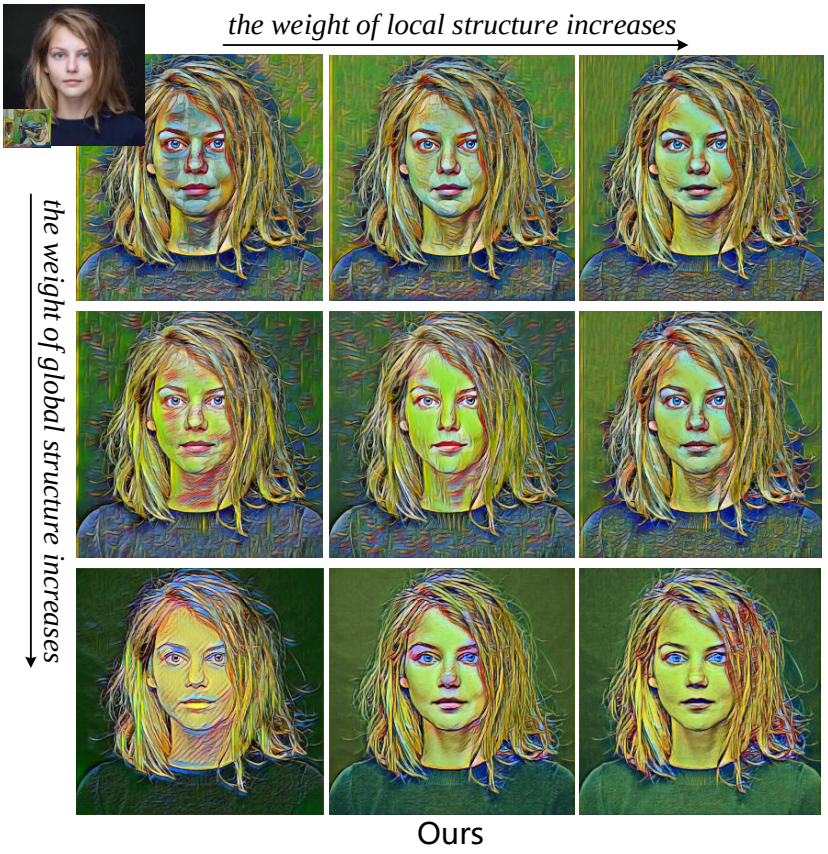


Fig. 8. The results under various combinations of local and global structures. The overall tendency is: as the weighting for local structure increases, the face becomes cleaner and smoother, with the amount of clutter edges continually reducing. As the weight of global structure increases, the foreground is increasingly apparent.

structures capture this trait well. In this way, we maintain the appearance of the portrait while encouraging the application of artistic effects. In contrast, in the results of [17], [20], densely covered textures make the human face untidy and even disfigure it. For the same reason, when processing other images, like the gondolas (See Fig. 5), although they all belong to the background, the sky and the sea are treated differently: the curling waves are transferred to different patterns, but the sky remains clear. Meanwhile, we note that the emphasis on structures will naturally cause the loss of abstract feelings. And this can be seen when compared with traditional methods (for example as shown in Fig. 7). We further discuss this limitation in Sec. V-E.

We also compare our method with some photorealistic style transfer methods (Fig. 6). Among them, [32], [34] are designed for the special cases that the content and the style image share highly similar layout and semantic components. The results show that our method can also work well in such cases.

All the above are due to the use of the structure enhanced image representation, in which two components play different roles: 1) Global structures are helpful for holding and preserving the overall layout; 2) Local structures mainly focus on the local information and well capture the minute features, and thus can refine local details on the basis of the global structures. These two kinds of structures supplement each other, and neither can perform effectively without the other. It is their interactive effects that promote the visual quality.

Sensitivity Analysis on Structure-Reconstruction Loss. The structure reconstruction loss $l_{structure}$ includes two parts: local-structure loss and global-structure loss. Their different combinations are responsible for different artistic effects.

We have tried several structure losses and show the results in Fig. 8. The weight of l_{edge} (α) increases from left to right, the weight of l_{depth} (β) increases from top to bottom. We can find that: 1) As α increases, the face and the background are becoming more clean and smooth, and the amount of clutter edges reduces; 2) With the increase of β, the foreground becomes increasingly more apparent. When α and β are in proper proportions, we are able to get very nice effects.

In short, our method provides an adjustable way to have a better control when stylizing images. Structure enhanced losses with higher share of l_{depth} are suitable for landscapes and close-up shots; those with a greater weight of l_{edge} are appropriate for portraits.

Ablation Analysis. We show ablation analyses in Fig. 9. Since we have made network sacrifices for the sake of ensuring processing speed, it is unsurprising that the results of the baseline model, whose image representation contains no structure component, are not so good. And this just reflects the power of the structure enhanced representation.

If incorporating only the local-structure (see the second

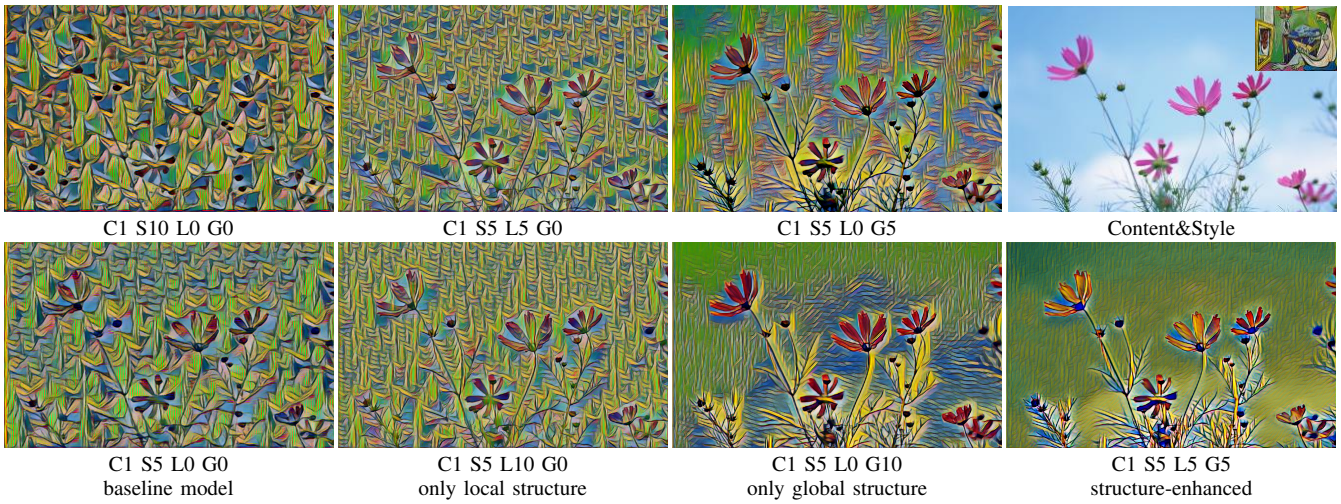


Fig. 9. Ablation analyses (**C**:content **S**:style **L**:local-structure **G**:global-structure, the numbers are the weights). The first column is the result of baseline model that uses no structure component in the image representation. The second column is the result only using local structure. The third column only uses global structure. The final column is the result using both local and global structures.

column), the local clutter is decreased but the overall feeling is still messy. If incorporating only the global-structure (see the third column), the overall structure of space is enhanced but some local parts still need to be addressed. Combining both local and global structures improves the results.

Speed and Memory Analysis. We compare the runtime of our method and [10], [17], [20], [41] for several image sizes in Table I. On the whole, the speed of our method is the fastest, making sure it can run in real-time or on videos. Table II shows the memory required for stylization of a single image of size 768×768 pix for different methods.

Methods	Image Size		
	256×256	512×512	1024×1024
Gatys *et al*. [10] (500 iterations)	15.86s	54.85s	214.44s
Johnson *et al*. [20]	0.015s	0.05s	0.21s
Ulyanov *et al*. [41]	0.021s	0.046s	0.145s
AdaIN [17]	0.018s	0.065s	0.275s
Ours	**0.008s**	**0.023s**	**0.12s**

TABLE I
SPEED COMPARISON. THE SPEED OF [10] IS SLOW, [17], [20] CAN NEARLY RUN IN REAL-TIME ON 512×512 IMAGES. COMPARED WITH THEM, OUR METHOD IS QUICKER. ALL THE RESULTS ARE OBTAINED WITH A TITAN X 12GB GPU.

Methods	[10]	[20]	[17]	Ours
GPU memory	3380 MiB	665 MiB	8869 MiB	**502 MiB**

TABLE II
AVERAGE GPU MEMORY CONSUMPTION, MEASURED ON A TITAN X GPU, FOR DIFFERENT METHODS WITH BATCH SIZE 1 AND INPUT IMAGE OF 768×768 PIX.

User Studies. We conduct a user study to evaluate the performance since the aesthetic evaluation of image stylization is a highly subjective task. We use the Sojump online questionnaire and voting platform. The respondents include two groups: 108 pupils under 12, and 112 adults aged between 18 and 30. Every questionnaire randomly selects 9 pairs from a set of 25 content-style pairs, and participants are asked to vote for their favorite stylized results (all the participants) and results with greater structural consistency (adults only, because we found that the majority of pupils have no clear conception of this). We compute the selected percentages of every algorithm and regard them as the preference scores.

Table III compares our algorithm with other artistic stylization algorithms for user preference scores. The survey results of pupils and adults are similar overall. We find the proposed algorithm is preferred for its stylization effect, and adult participants reach a consensus that our method better retains structure consistency.

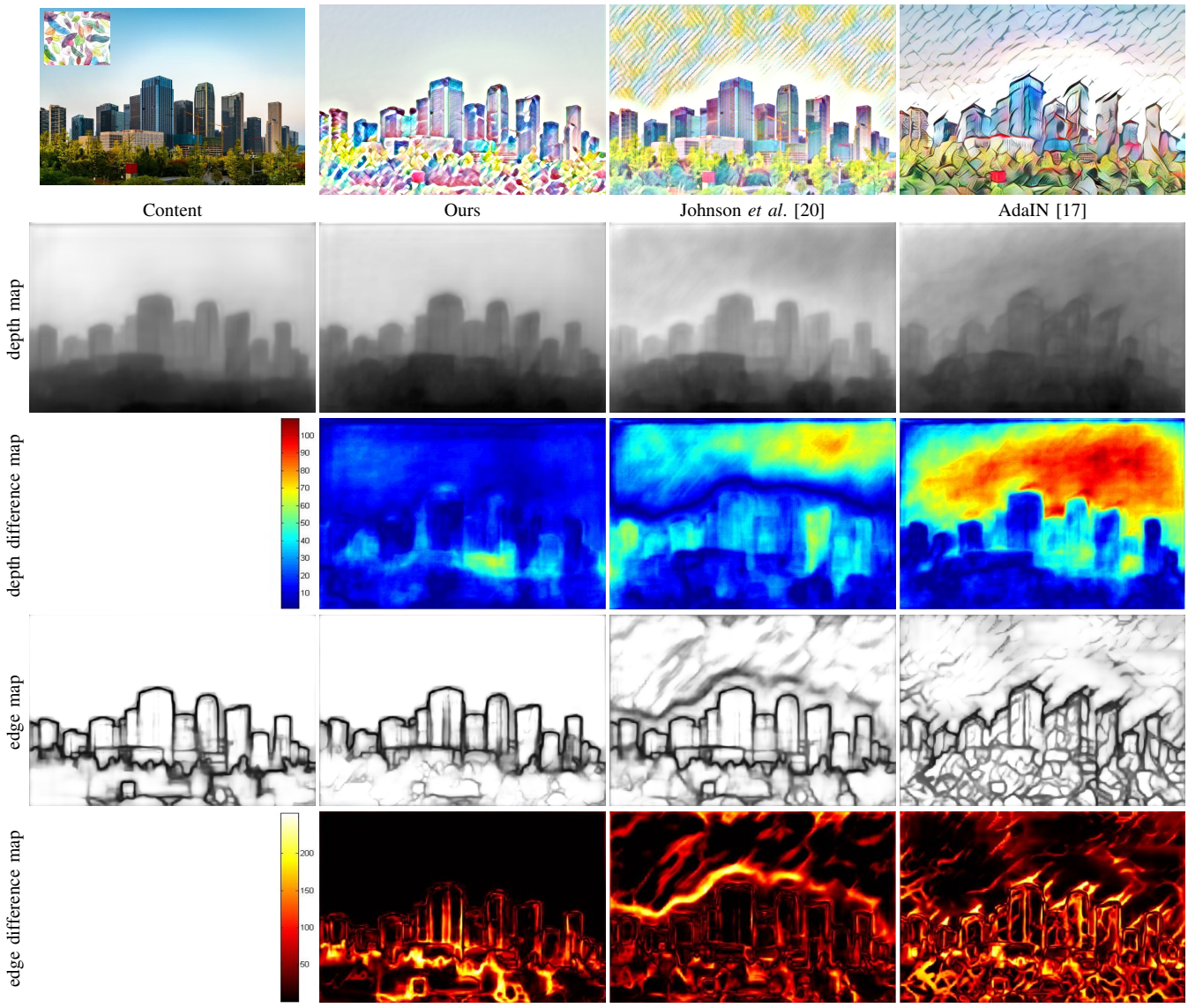
Methods	Favorite Stylization			Structural
	Pupils	Adults	Overall	Consistency
Gatys *et al*. [10]	19.14%	12.70%	15.86%	12.81%
Johnson *et al*. [20]	20.99%	18.65%	19.80%	11.88%
AdaIN [17]	17.90%	21,83%	19.90%	10.00%
Ours	**41.98%**	**46.83%**	**44.44%**	**65.31%**

TABLE III
USER PREFERENCE

C. Analysis on the structure consistency.

Instead of keeping accurate structures, our system was designed to maintain the structural consistency between the original images and transferred images. This was achieved under the guidance of the global structure extraction network and local structure refinement network. In the training stage, these networks give feedback based on the stylization performance.

In addition to the visual feeling, it is indeed interesting to further check whether the proposed network keeps structure consistency. Currently our results keep plausible depth and edge maps with consistent spatial and structure senses. By comparing the depth and edge maps computed from the content and stylized images respectively, as shown in Fig. 10, we find that our results can better recover the maps than other methods. This can be illustrated more intuitively by the depth difference and edge difference maps. Moreover, we use SSIM [43] and RMSE to measure the similarity between



Methods	Content	Depth Map		Edge Map	
	SSIM	SSIM	RMSE	SSIM	RMSE
Ours	**0.629**	**0.956**	**16.716**	**0.808**	**43.656**
Johnson *et al.* [20]	0.392	0.939	33.141	0.755	50.576
AdaIN *et al.* [17]	0.394	0.895	53.906	0.582	56.813

Fig. 10. Comparison on the structure consistency. The second row is the depth prediction results using [3], the third row is the edge detection results using [31]. The second and the last row are the corresponding difference map. The results indicate that we recover the overall depth structure of the scene quite well. And the results of the others introduce some tiny edges, which spread over the whole image. The table blow shows the metric error measures (SSIM [43], RMSE) on the results. Higher is better for SSIM (structural similarity), lower is better for RMSE (root mean squared error).

the resulting image and the original. The SSIM is an index measuring the structural similarity between two images. When two images are nearly identical, their SSIM is close to 1. RMSE (the root mean squared error) computes the absolute difference of two images. We see that under these metric measures, our stylized result, the depth map and the edge map preserve the structure consistency well, which demonstrates the effectiveness of our schemes.

D. Understanding Structure-Representation Network

We want to explain the motivation and reason for designing the Structure-Representation Network here:

1) Can existing methods achieve an equal effect by adjusting the balance between l_{style} and $l_{content}$?

Some might wonder if we could get similar results by increasing the weight of $l_{content}$. Although this sounds reasonable at first, it is not the case.

First, we should note that $l_{content}$ is computed as the distance of middle layer features in the VGG network, and VGG was designed for object recognition. So these features will concentrate mainly on the primary target, the backgrounds and other objects cannot be fully represented. Next, even in one object, the structures may have many variations, such as the local edge details in a portrait (rich in hair and less in face) or depth values in a road (changing with the extension

(a) Ours (b) Johnson *et al*. [20]

Fig. 11. Can existing methods achieve an equal effect by increasing the weight of $l_{content}$? The right two columns are the results of [20]: The weight of $l_{content}$ increases from left to right, up to down. We can see that simply increasing the weight of $l_{content}$ will only make results increasingly like the content image, the styles are still evenly distributed in the results.

of the road). Hence we need to introduce some other structure losses to preserve these details.

The experiments further validate our analysis. As Fig. 11 shows, existing methods cannot achieve structure-enhanced stylized results by simply adjusting the weight of $l_{content}$.

2) Why choose the local and global structures?

When analyzing the results of existing methods, as shown in Figs. 2 and 5, we first notice that the even distribution of style textures is a universal problem, and this makes it hard to distinguish between the foreground and background. We describe this phenomenon as the lack of global structure. Considering that the depth map effectively reflects the spatial distribution in an image, we think that if we can preserve the depth information of the content image after stylization, better results could be obtained.

After preserving the global structure, as shown in Fig. 9, the problem mentioned previously has truly been improved. We can see the overall structure was enhanced, but style textures are still uniformly distributed in the areas where the depth values are roughly the same (such as the background of Fig. 9). So it is not enough to solve the problems if we only consider the overall structures.

To compensate for the limitations of the global structure, we add the local-structure refinement network because in human visual perception an excessive amount of style textures will indirectly introduce extra edges and break the local structure. So if we force the local structure of stylized images to be consistent with the content image to some extent, then the above phenomenon will be eliminated, as shown in Fig. 9.

Therefore, we finally design the structure representation network to improve the visual quality. The degree to which these two structures affect the visual quality was discussed in Sec. V-B.

E. Limitations

On the whole, our method provides an adjustable way to better retain or enhance structure when stylizing images. Since everyone has their own preferences and every image has its own characteristics, our method may not please everybody or be suitable for all images. If you prefer an abstract feeling and the content images are not sensitive to structural changes, then techniques such as Gatys *et al*. [10] may be more effective (see Fig. 12).

(a) Content (b) Gatys *et al*. [10] (c) Ours

Fig. 12. Limitations of our method. Our method is more suitable to process images which are sensitive to structural changes. For images, like this cat, the deformation of the structures and evenly distributed styles in Gatys *et al*. [10] actually give it an abstract feeling which could be considered attractive. Our result well retain the structures but loses the abstraction.

VI. Conclusion

In this paper we propose an approach for image stylization in which structures are preserved and enhanced. Under the guidance of the global structure extraction network and local structure refinement network, we successfully retain layout structures while applying artistic effects. Experimental results demonstrate that our method achieves an impressive visual effectiveness, which is particularly significant when processing images which are sensitive to structure distortion. The experimental results also confirm that image representation plays a very important role in stylization, and different stylization results can be generated by constructing alternative image representation strategies.

Acknowledgements. This research was supported by NSFC (NO. 61572264), the national youth talent support program, and Tianjin Natural Science Foundation (17JCJQJC43700).

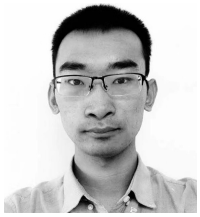
References

[1] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. StyleBank: an explicit representation for neural image style transfer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[2] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. In *Adv. Neural Inform. Process. Syst. Worksh.*, 2016.
[3] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Adv. Neural Inform. Process. Syst.*, 2016.
[4] V. Dumoulin, J. Shlens, M. Kudlur, A. Behboodi, F. Lemic, A. Wolisz, M. Molinaro, C. Hirche, M. Hayashi, E. Bagan, et al. A learned representation for artistic style. In *Int. Conf. Learn. Represent.*, 2017.
[5] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH*, 2001.
[6] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Int. Conf. Comput. Vis.*, 1999.
[7] L. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, 2015.
[8] L. A. Gatys, M. Bethge, A. Hertzmann, and E. Shechtman. Preserving color in neural artistic style transfer. `arXiv:1606.05897 [cs.CV]`, 2016.
[9] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. `arXiv:1508.06576 [cs.CV]`, 2015.
[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.
[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Adv. Neural Inform. Process. Syst.*, pages 2672–2680, 2014.

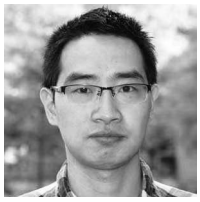
[12] A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei. Characterizing and improving stability in neural style transfer. In *Int. Conf. Comput. Vis.*, 2017.
[13] A. Hausner. Simulating decorative mosaics. In *ACM SIGGRAPH*, 2001.
[14] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *ACM SIGGRAPH*, 1995.
[15] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *ACM SIGGRAPH*, 2001.
[16] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr. Deeply supervised salient object detection with short connections. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(4):815–828, 2019.
[17] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Int. Conf. Comput. Vis.*, 2017.
[18] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size. `preprint arXiv:1602.07360 [cs.CV]`, 2016.
[19] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song. Neural style transfer: A review. *IEEE Trans. Vis. Comput Graph.*, 2019.
[20] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Eur. Conf. Comput. Vis.*, 2016.
[21] B. Julesz. Visual pattern discrimination. *IRE Transactions on Information Theory*, 8(2):84–92, 1962.
[22] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent.*, 2014.
[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, 2012.
[24] C. Li and M. Wand. Combining Markov random fields and convolutional neural networks for image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.
[25] C. Li and M. Wand. Precomputed real-time texture synthesis with Markovian generative adversarial networks. In *Eur. Conf. Comput. Vis.*, 2016.
[26] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Diversified texture synthesis with feed-forward networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[27] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz. A closed-form solution to photorealistic image stylization. In *Eur. Conf. Comput. Vis.*, 2018.
[28] Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying neural style transfer. In *Int. Joint Conf. Artifi. Intell.*, 2017.
[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Eur. Conf. Comput. Vis.*, 2014.
[30] X.-C. Liu, M.-M. Cheng, Y.-K. Lai, and P. L. Rosin. Depth-aware neural style transfer. In *Non-Photorealistic Animation and Rendering (NPAR)*, pages 4:1–4:10, 2017.
[31] Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bian, L. Zhang, X. Bai, and J. Tang. Richer convolutional features for edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.
[32] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[33] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.
[34] R. Mechrez, E. Shechtman, and L. Zelnik-Manor. Photorealistic style transfer with screened Poisson equation. In *Brit. Mach. Vis. Conf.*, 2017.
[35] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. J. Comput. Vis.*, 40(1):49–70, 2000.
[36] L. Ruizhi, X. Yu, and Z. Xiuming. Depth-preserving style transfer. 2017.
[37] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Int. Conf. Learn. Represent.*, 2014.
[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent.*, 2015.
[39] X. Snelgrove. High-resolution multi-scale neural texture synthesis. In *SIGGRAPH ASIA Technical Briefs*. ACM, 2017.
[40] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1226–1238, 2002.
[41] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. S. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Int. Conf. Mach. Learn.*, 2016.
[42] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.
[44] P. Wilmot, E. Risser, and C. Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. `arXiv:1701.08893 [cs.GR]`, 2017.
[45] S. Xie and Z. Tu. Holistically-nested edge detection. *Int. J. Comput. Vis.*, 2017.
[46] J. Xu and C. S. Kaplan. Calligraphic packing. In *Proceedings of Graphics Interface*, pages 43–50. ACM, 2007.
[47] H. Zhang and K. Dana. Multi-style generative network for real-time transfer. `arXiv:1703.06953 [cs.CV]`, 2017.
[48] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Int. Conf. Comput. Vis.*, pages 2242–2251, 2017.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did 2 years research fellow, with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests includes computer graphics, computer vision, and image processing. He received research awards including ACM China Rising Star Award, IBM Global SUR Award, CCF-Intel Young Faculty Researcher Program, etc.

Xiao-Chang Liu is currently a Master student with the College of Computer Science, Nankai University, under the supervision of Prof. Ming-Ming Cheng. His research interests include image processing and computer vision.



Jie Wang is currently an Undergraduate student with the College of Computer Science, Nankai University. His research interests include deep learning and computer vision.



Shao-Ping Lu received the Ph.D. degree in Computer Science at Tsinghua University, China, in July 2012. From November 2012, he has been a Postdoctoral Researcher at Vrije Universiteit Brussels (VUB) in Belgium, as a senior researcher. He is now an associate professor at Nankai University. His primary research areas are image & video processing and editing.

Yu-Kun Lai received his bachelor's and Ph.D. degrees in computer science from Tsinghua University, China, in 2003 and 2008, respectively. He is currently a reader at the School of Computer Science & Informatics, Cardiff University. His research interests include Computer Graphics, Computer Vision, Geometry Processing and Image Processing.

Paul L. Rosin is a professor at the School of Computer Science & Informatics, Cardiff University. His research interests include the representation, segmentation, and grouping of curves, knowledge-based vision systems, early image representations, low level image processing, machine vision approaches to remote sensing, methods for evaluation of approximation algorithms, medical and biological image analysis, mesh processing, non-photorealistic rendering and the analysis of shape in art and architecture.