Fine-Grained Image Style Transfer with Visual Transformers

Jianbo Wang[1], Huan Yang[2], Jianlong Fu[2],
Toshihiko Yamasaki[1], and Baining Guo[2]

[1] The Univeristy of Tokyo {jianbowang815,yamasaki}@cvm.t.u-tokyo.ac.jp
[2] Microsoft Research {huayan,jianf,bainguo}@microsoft.com

1 Architecture of STTR's Transformer

1.1 Tokenizer

In our model, one image is divided into a set of visual tokens. Thus, we have to first convert the input image into a set of visual tokens. We assume that each of them represents a semantic concept in the image. We then feed these tokens to a transformer. Let us denote the input feature map by $X \in \mathbb{R}^{H\times W\times C}$ (height H, width W, and channels C) and visual tokens by $T \in \mathbb{R}^{L\times C}$ where L indicates the number of tokens.

In the main paper, we adopt a filter-based tokenizer. Here we would like to compare the two tokenizers (i.e., unfold-based tokenizer and filter-based tokenizer) and explain why we choose the filter-based tokenizer.

Unfold-based Tokenizer. The first choice is to unfold the features by using a sliding window to slide along H and W dimensions to create blocks (always with overlaps). Let us assume the output size of extracted feature maps is $H \times W \times C_1$, where H, W, and C_1 indicate height, width, and the number of channels. As shown in Fig. 1(a), we directly slide local blocks from image features into patches. If the shape of each patch is $h \times w$ (height of each patch h, width of each patch w), then we could obtain L tokens. For each token, the dimension is $h \times w \times C_1$. L could be calculated as follows:

$$L = \prod_{d} \left\lfloor \frac{spatial\_size[d] - kernel\_size[d]}{stride[d]} + 1 \right\rfloor , \tag{1}$$

where d is overall spatial dimensions, *spatial_size*, *kernel_size*, and *stride* are formed by the spatial dimensions of input, the size of the convolution kernels, and the stride for the sliding window. $\lfloor x \rfloor$ function calculates the largest integer that is less than or equal to x. Usually, $L \ll H \times W$. Specific to our model, such a design may result in strong blocky artifacts. This is because we reshape the feature from the spatial dimension into the channel dimension and the style loss will constrain the predicted results to match the mean and variance with the target style features in each channel.

Transformer Encoder The detailed architecture of STTR's Transformer could be found in Fig. 2.

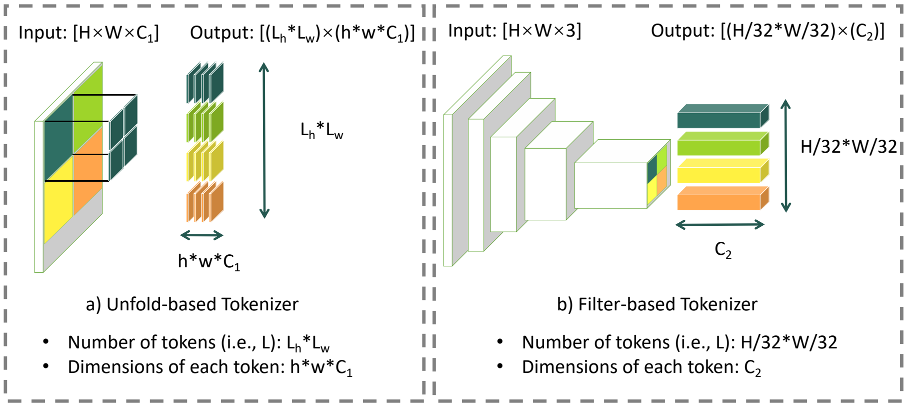

Input: [H×W×C$_1$]
Output: [(L$_h$*L$_w$)×(h*w*C$_1$)]
L$_h$*L$_w$
h*w*C$_1$
a) Unfold-based Tokenizer
• Number of tokens (i.e., L): L$_h$*L$_w$
• Dimensions of each token: h*w*C$_1$
Input: [H×W×3]
Output: [(H/32*W/32)×(C$_2$)]
H/32*W/32
C$_2$
b) Filter-based Tokenizer
• Number of tokens (i.e., L): H/32*W/32
• Dimensions of each token: C$_2$

Fig. 1. Illustration of different dimension configurations of the backbone. (a) Unfold-based tokenizer maintains spatial dimensions of features and then unfolds them into patches; (b) Filter-based tokenizer gradually downsamples the features using ResNet-50 to obtain spatially smaller output features.

2 Ablations

The Tokenizer We compare the outputs from different dimension configurations of the backbone. As shown in Fig. 4(a), the result from the unfold-based tokenizer has strong blocky artifacts. This is because the model only sees patches and could thus create artifacts on the borders of these patches. Also, the supervision signal from the style loss will guide the predicted results to match the mean and variance with the target style features in each channel. This will strengthen the artifacts because we have reshaped the feature from the spatial dimension into the channel dimension. In contrast, the filter-based tokenizer produces features in a much more smooth manner and outputs visual pleasant results (see Fig. 4(b)). More details can be found in the zoom-in view in Fig. 4.

2.1 More ablation studies for each components

Number of Encoder and Decoder Layers In Fig. 3, we show stylized results generated by our model with different layer numbers of encoder and decoder. We can observe model with deeper encoder and decoder has stronger capability to preserve semantic similarity, so that similar style patterns (e.g., fire) can be transferred to similar content regions.

The Receptive Field in the Backbone In Fig. 5, we change the depth of content and style backbone independently. The receptive field size in the backbone could affect the stroke size. The larger receptive field for extracting content features could make the final results miss details and also spend more time converging (see the content backbone with four layers in Fig. 5). While for extracting

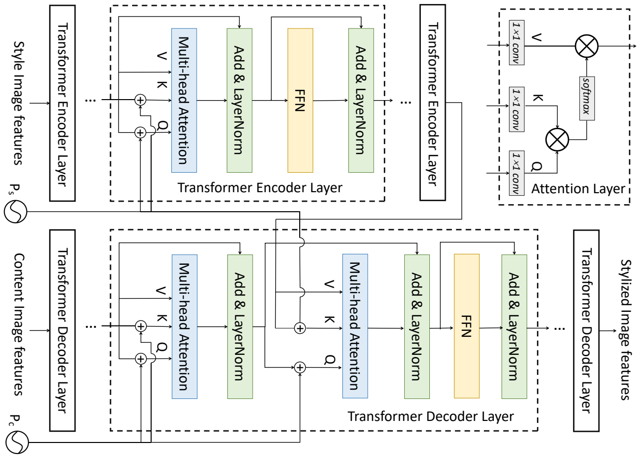


Fig. 2. Architecture of STTR's Transformer, including Transformer encoder and decoder. The ⊕ and ⊗ represent matrix addition and dot-product operations, respectively. P_s and P_c represent the positional encoding of style and content features, respectively.

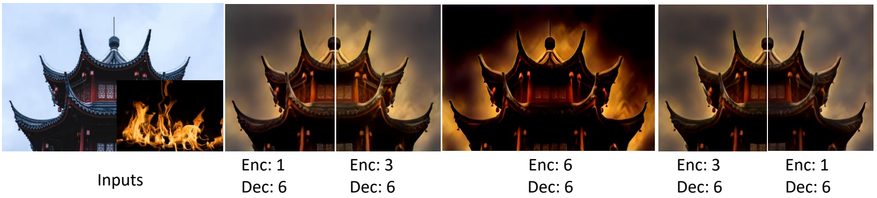


Fig. 3. Effects of different encoder and decoder sizes. "Eec" and "Dec" indicate the number of encoder and decoder layer respectively.

style features, the larger receptive field always appears with a deeper backbone which could extract higher-level semantic features for better understanding the style pattern. For example, in Fig. 5, the results with deeper style backbone (4 layers) show much more smooth content. Thus, for style features, we suggest using a deeper backbone while for content features, a shallow backbone is recommended. This is also consistent with the observation in [1].

In the experiments, we have also tried the Gram matrix loss [2] to replace the AdaIN style loss [3] as presented in Sec. 3.2 in the main paper. However, the results show that the AdaIN style loss performs better.

The Receptive Field in the loss network Since features from different layers capture different style details. The degree of stylization can be modified by using multiple levels of features. As shown in Fig. 6, transferring content and style features from shallower layers ($relu1\_1$) produce more photo-realistic visual

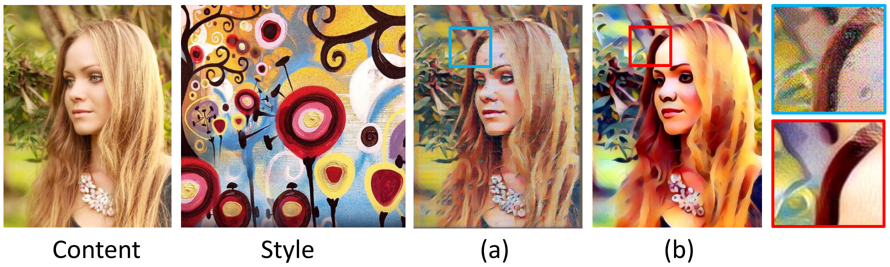


Fig. 4. Visualization of outputs from different dimension configurations of the backbone. (a) Unfold-based tokenizer; (b) Filter-based tokenizer. Zoom in for a better view.

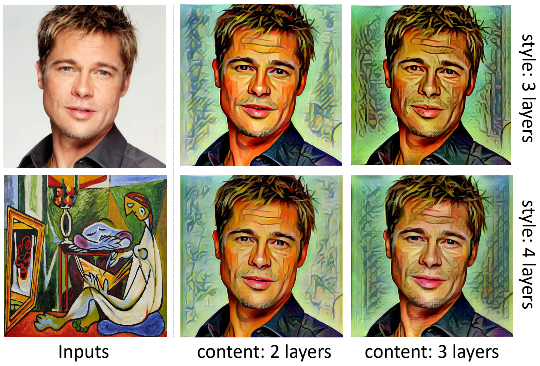


Fig. 5. Visual comparison with different receptive field in the content and style backbone.

effects. On the other hand, using features from deeper layers bring more abstract style to the sky. Optimal results could be obtained by computing the average value of feature difference in the four layers.

3 Control the Style Size

As style transfer is a very subjective task, sometimes we require realistic results (preserving more details) while sometimes we prefer artistic stylization (more abstract style). We could control the magnitude of stylization by the following three factors in our proposed STTR:

- The receptive field in the loss network (Sec. 3.2 in the main paper and Sec. 2.1).
- The receptive field in the backbone (Sec. 3.1 in the main paper and Sec. 2.1).
- The loss weight λ (Sec. 3.2 in the main paper and Sec. 4.3 in the main paper).

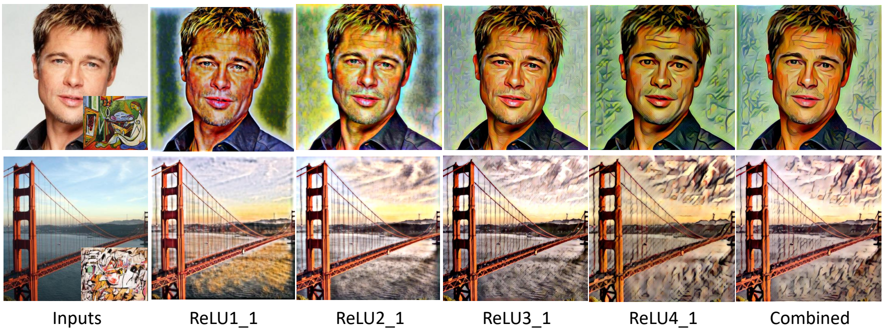


Fig. 6. Results with losses at different levels. We use a fixed-weight VGG-19 as our loss network.

For Fig. 3 in the main paper, we set the content backbone with activation after 2 layers while style backbone with 4 layers, $\lambda = 10$. During training and testing, increasing the receptive field of the style backbones or the loss network, or setting a larger λ, could provide richer styles.

4 Additional Experiments

4.1 More Qualitative Results for Image Style Transfer

We provide further comparisons obtained by our proposed STTR and other state-of-the-art methods. We evaluate various content images with distinctive styles. The results are illustrated in Fig. 7. The results of compared methods are obtained by running their codes with default configurations. All of the images used in the testing stage are never observed during the training stage.

5 Details of User Study

To evaluate the results, we conduct a user study on Amazon Mechanical Turk (AMT). We have designed a website to show comparison results between other state-of-the-art methods and our proposed STTR. We use 10 content images and 30 style images collected from copyright-free websites. For each method, we use the released codes and default parameters to generate 300 content-style pairs. We hire 75 volunteers on Amazon Mechanical Turk (AMT) for our user study. 20 of 300 content-style pairs are randomly selected for each user. The screenshot of the designed website is shown in Fig. 8.

The instruction for this questionnaire is as follows:

Which one is the most visually pleasant, maintaining the content structure well with the given content image and along with the given style pattern?

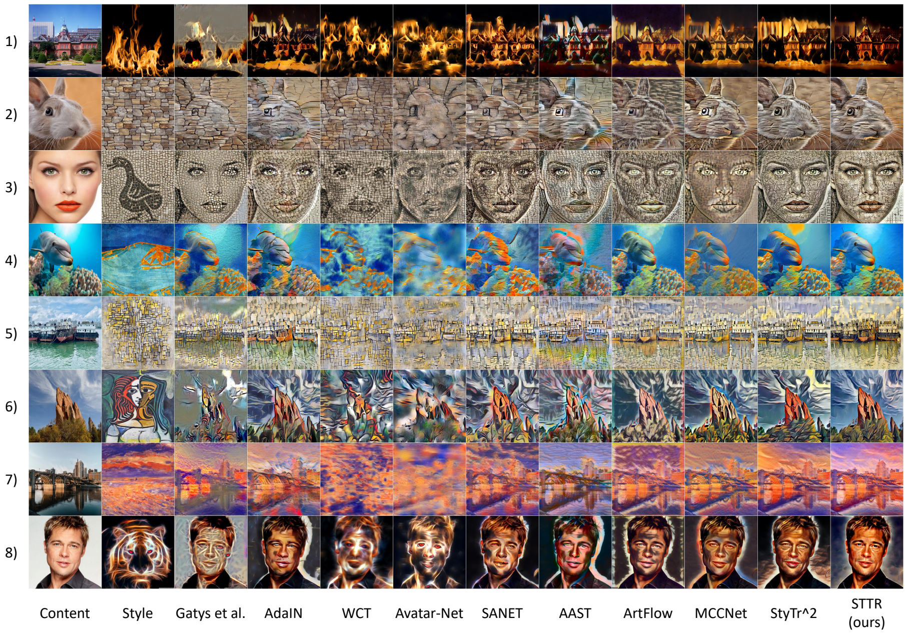
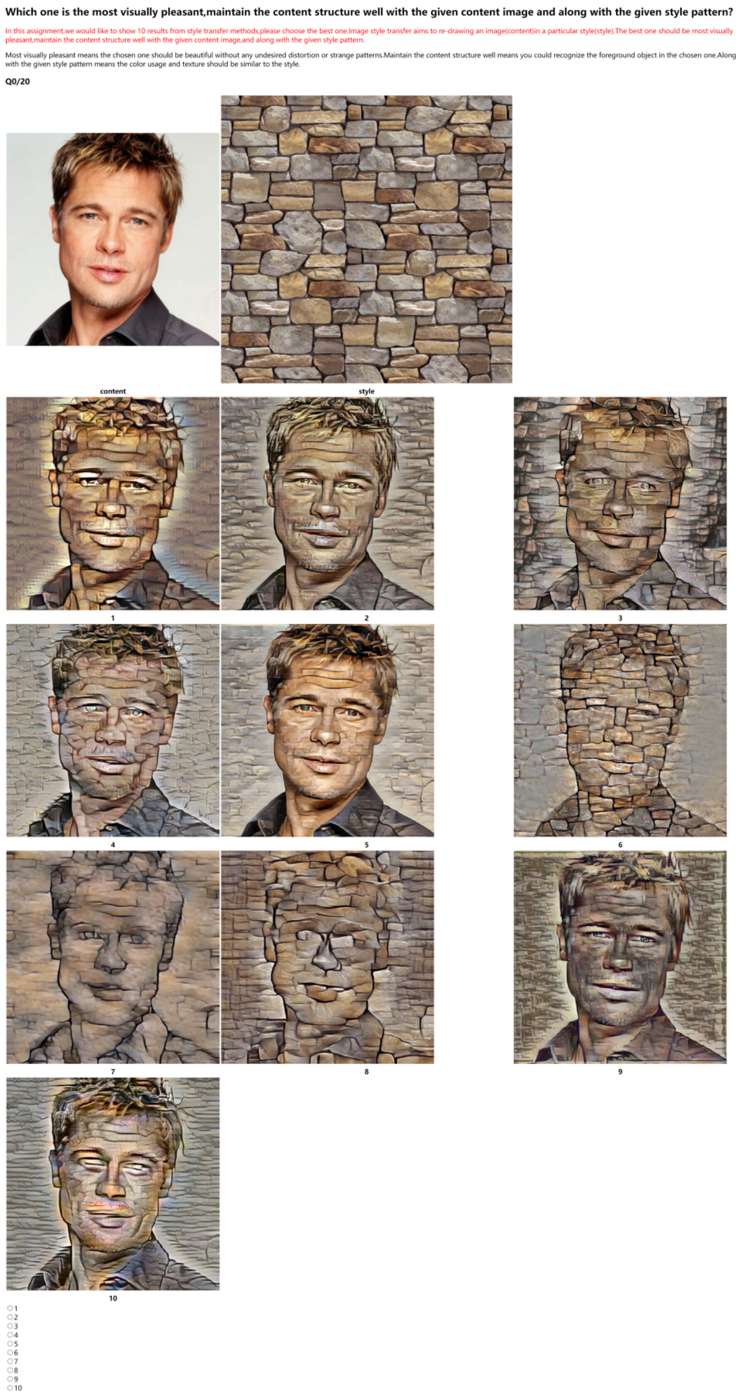


Fig. 7. More Qualitative Results for Image Style Transfer

In this assignment, we would like to show seven results from style transfer methods, please choose the best one. Image style transfer aims to re-drawing an image (content) in a particular style (style). The best one should be most visually pleasant, maintain the content structure well with the given content image, and along with the given style pattern.

Most visually pleasant means the chosen one should be beautiful without any undesired distortion or strange patterns. Maintaining the content structure well means you could recognize the foreground object in the chosen one. Along with the given style pattern means the color usage and texture should be similar to the style.

Each user is asked to vote for only one result (they don't know which one is generated by which method). The average time for a volunteer to finish the questionnaire is five minutes. Finally, we collect 1500 votes from 75 users and calculate the percentage of votes that each method received.



Which one is the most visually pleasant,maintain the content structure well with the given content image and along with the given style pattern?

In this assignment,we would like to show 10 results from style transfer methods,please choose the best one.Image style transfer aims to re-drawing an image(content)in a particular style(style).The best one should be most visually pleasant,maintain the content structure well with the given content image,and along with the given style pattern.

Most visually pleasant means the chosen one should be beautiful without any undesired distortion or strange patterns.Maintain the content structure well means you could recognize the foreground object in the chosen one.Along with the given style pattern means the color usage and texture should be similar to the style.

Q0/20

○1
○2
○3
○4
○5
○6
○7
○8
○9
○10

Fig. 8. Questionnaire for comparison between other state-of-the-art methods in the user study.

References

1. Jing, Y., Liu, Y., Yang, Y., Feng, Z., Yu, Y., Tao, D., Song, M.: Stroke Controllable Fast Style Transfer with Adaptive Receptive Fields. In: ECCV. (2018) 238–254
2. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR. (2016) 2414–2423
3. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: ICCV. (2017) 1501–1510