Improving the Latent Space of Image Style Transfer

Anonymous authors
Paper under double-blind review

Abstract

Existing neural style transfer studies utilize statistical information of features from a pre-trained encoder as representations of the style and achieve significant improvement in synthesizing artistic images. However, in some cases, the feature statistics from the pre-trained encoder may not be consistent with the visual style we perceived. The style distance between some images of different styles is small than that of the same style. In such an inappropriate latent space, the objective function of the existing methods will be optimized in the wrong direction, resulting in bad stylization results. In addition, the lack of content details in the features extracted by the pre-trained encoder also leads to the content leak problem. In order to solve these issues in the latent space used by style transfer, we propose two contrastive training schemes to get a refined encoder that is more suitable for this task. The style contrastive loss pulls the stylized result closer to the same visual style image and pushes it away from the content image. The content contrastive loss enables the encoder to retain more available details. The training scheme can be directly added to existing style transfer methods and significantly improve their results. Extensive experimental results demonstrate the effectiveness and superiority of our methods.

1 Introduction

Artistic style transfer Gatys et al. (2016); Li et al. (2017); Park & Lee (2019); Chen et al. (2021); Li et al. (2019); Sheng et al. (2018) has been a long-term research topic that aims to transfer artistic style from reference image to content image. Recent methods Gatys et al. (2016); Huang & Belongie (2017) use neural networks to match feature statistical information between content and style images. Although these approaches have developed rapidly and achieved significant improvement, there remains a critical problem that has not been discussed: *Is the style forms used by the existing methods, which are based on feature statistics, consistent with the characteristics of visual styles?*

Gatys et al. Gatys et al. (2016) first proposed the neural style transfer method, which uses image representations derived from a pre-trained Deep Convolutional Neural Network (DCNN) to separate image content from style. In their way, style is defined as the Gram matrix of the deep features, which is related to the fixed parameters of the encoder. Different style definitions are proposed in the later work, but they mostly describe a pre-trained encoder's deep feature statistics (e.g., mean and variance of features Huang & Belongie (2017)). The encoder pre-trained in different ways get different style representation values for the same image. *So, which encoder is more suitable for style transfer?* Some works Du (2020); Wang et al. (2021a) have pointed out that even the randomly initialized network can also achieve acceptable style transfer results. This shows that the encoder pre-trained on large collections of images is unnecessary, and it may not even be the most suitable for style transfer.

On the other hand, we judge the style of images through subjective perception, which may not be consistent with the feature statistics obtained by a pre-trained neural network. As the example shown in Figure 1, we use the method in Wang et al. (2020) to calculate the style distance between images with Gram matrix:

$$\mathcal{D}_{\text{style}} = \|\mathcal{G}(\mathcal{F}(I_1)) - \mathcal{G}(\mathcal{F}(I_2))\|_2. \tag{1}$$

This Gram matrix is obtained by a commonly used pre-trained VGG-19 Simonyan & Zisserman (2015). It can be seen that the style distance between different style images is smaller than that of the

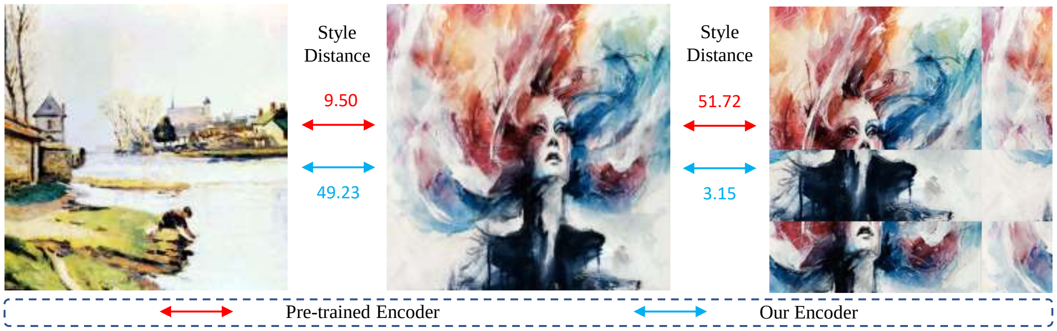


Figure 1: The two images on the left are randomly selected from the style dataset. We rearrange one of them spatially to get the third image. Then, We use a pre-trained encoder used in previous methods to calculate the style distance for three images. Surprisingly, we find that the distance between the two images on the left was smaller than that of the two images with exactly the same style on the right, which indicates that the objective of the existing methods may not be consistent with the visual style.

same style images in the feature space of the pre-trained encoder. Even if we optimize the style loss of existing methods to a smaller value in such an inappropriate latent space, the stylized result may not achieve a visually consistent style. Therefore, we consider optimizing the encoder's parameters while training other modules to make the style representations in its latent space consistent with characteristics of visual styles. Wang et al. Wang et al. (2020) try to use the knowledge distillation method to get a new encoder from the pre-trained encoder and achieve better results. However, the new encoder retains the same knowledge as the original encoder in their training process. That is, its latent space has not changed much.

In order to solve this problem, we design a style contrastive training scheme to fine-tune the pre-trained encoder to get a more suitable one, which features statistics that are more style-consistent. By pulling the style positive examples and pushing away the negative examples, the same visual style images will have a more consistent representation in the latent space of the encoder. This training scheme can be directly added to the existing style transfer methods and improve the effect.

In addition, there is another problem with existing style transfer methods: The stylization results of these CNN-based methods often lose some content information An et al. (2021). Some works Park & Lee (2019) try to solve this problem, but they can only train the decoder part to retain more content at most, while ignoring the missing content details in the features extracted by the encoder. For this, we also propose an identity preserve content contrastive loss to make the encoder retain more local details in fine-tuning. Finally, we conduct experiments on some state-of-the-art style transfer methods and achieved significant improvement.

To summarize, the main contributions of this work are threefold:

- We propose a style contrastive training scheme to refine the pre-trained encoder used in the existing style transfer method to make its latent space more style-consistent.
- We propose an identity preserve content contrastive loss to alleviate content leak problem caused by the pre-trained encoder.
- We demonstrate the effectiveness and superiority of our approach by adding our training scheme to some existing methods and achieve significant improvement.

2 Related Work

2.1 Style Transfer

Artistic style transfer aims to transfer the style of some artworks to real-world photos, and create a large number of images that have not appeared before. Before the emergence of deep neural network, similar tasks were more like a problem of texture transfer, which mainly tackled by non-parametric sampling Efros & Leung (1999), non-photorealistic rendering Gooch & Gooch (2001); Strothotte & Schlechtweg (2002) or image analogy Hertzmann et al. (2001). With the help of DCNN, Gatys et al. Gatys et al. (2016) first propose the neural style transfer, which uses deep features from the pre-trained network to represent style and content. In their way, style is defined as the Gram matrix of deep features, and stylization is achieved by matching the second-order statistics between the result and style image. In the later work, different style definitions Huang & Belongie (2017) are proposed, while they mostly describe the statistics of deep features from a pre-trained encoder. However, these forms may not be consistent with the image's style, which is caused by the inappropriate encoder. Recently, some works have explored the influence of the encoder on style transfer, such as the encoder parameters Du (2020) and encoder architecture Wang et al. (2021a). Wang et al. Wang et al. (2020) try to use the knowledge distillation method to get a new encoder from the pre-trained encoder and achieve better results. However, the new encoder retains the same knowledge as the original encoder in their training process. That is, its latent space has not changed much. In contrast, we try to improve the latent space of the encoder to make it more style-consistent and apply the training scheme to the existing methods.

2.2 Contrastive Learning

Contrastive learning first appeared in the field of unsupervised representation learning and has shown great promiseHe et al. (2020). These methods are based on the theory of maximizing mutual information. The basic idea is to build an embedding space where associated signals are pulled together while other samples in the dataset are pushed away. Signals may vary depending on specific tasks. A new form of contrastive loss called InfoNCE van den Oord et al. (2018), which measures the similarity by dot production, is proposed as a representative loss function to maximize a lower bound of the mutual information. Later, the effectiveness of contrastive learning was gradually verified on various tasks, such as semantic segmentation Wang et al. (2021c), object detection Xie et al. (2021) and classification Wang et al. (2021b).

In the field of conditional image synthesis, contrastive learning has also received extensive attention Park et al. (2020); Yu et al. (2021). More recently, Liu et al. Liu et al. (2021) introduced a latent-augmented contrastive loss to achieve diverse image synthesis. Wu et al. Wu et al. (2021) improved the image dehazing result by pulling the restored image closer to the clear image and pushing it far away from the hazy image. Chen et al. Chen et al. (2021) first adapt contrastive learning to the artistic style transfer to learn so-called stylization-to-stylization relations. However, suppose the style transfer process is carried out in an unsuitable latent space. In that case, the stylization-to-stylization relations are meaningless, because they may all not be visually consistent with the style image. In contrast, we use the contrastive learning method to make the same visual style images have consistent representations in the latent space.

2.3 Knowledge Distillation

Knowledge distillation (KD) Ba & Caruana (2014); Hinton et al. (2015); Yu et al. (2019) is a model compression method, in which a student network is trained by learning the knowledge from a teacher network. The knowledge is expressed in the form of softened probability Yu et al. (2019); Peng et al. (2019), which can reflect the inherent class similarity structure known as dark knowledge. The distillation objective encourages the output probability distribution over predictions from the student and teacher networks to be similar. With the help of additional information on top of the one-hot labels, the performance of student network can be boosted. This dark knowledge is mainly related to labels, so they are rarely used in low-level vision tasks (e.g., neural style transfer). Wang et al. Wang et al. (2020) developed a collaborative knowledge distillation method to learn a much smaller model from pre-trained redundant VGG-19 for ultra-resolution style transfer. In our method, the pre-trained encoder is regarded as a regularizer to guarantee that the features extracted by the new encoder are near a suitable value.

3 Proposed Method

Arbitrary style transfer methods Huang & Belongie (2017); Park & Lee (2019); Chen et al. (2021) typically adopt an encoder-decoder architecture, which transfer the style in the encoder's feature space and convert them back to the stylized results through the decoder. The encoder often adopts

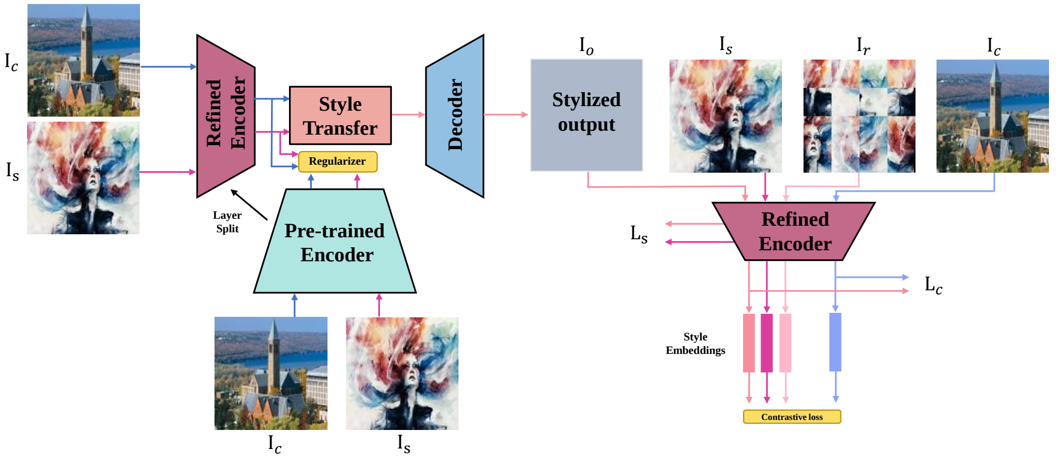


Figure 2: Our style contrastive training scheme. For some learning-based style transfer methods that use encoder-decoder architecture, we optimize their encoders in the training process simultaneously. The original pre-trained encoder is used as a feature regularizer. The stylized result is pulled closer to the same visual style image and pushed far away from the content image in the refined encoder's latent space. In the figure, We use similar colors to represent the embeddings of the same style. The style loss L_s and the content loss L_c remain the same as the original method.

a pre-trained VGG-19 Simonyan & Zisserman (2015) to extract expressive informative representations. In the training process, the encoder is used to provide a supervision signal, so its parameters cannot be optimized. However, the quality of the stylization results is affected due to the style inconsistency and lack of content details in the latent space of the pre-trained encoder. Therefore, we propose two training schemes to refine this encoder to make it more suitable for this task.

3.1 Encoder Fine Tuning

If the encoder's parameters are simply set to be optimized as other modules, it is equivalent to optimizing the output result and supervision signals simultaneously and will get degraded results. To avoid this problem, we use the original pre-trained encoder as a feature regularizer, which prevents the new features from being too far away from a suitable scale. The new encoder uses the same architecture and is initialized with the parameters of the pre-trained one. This practice makes the encoder converge more easily than random initialization. Further, we find that some layers in the original architecture are unnecessary. Therefore, only a few layers of the pre-trained encoder are retained in the new one. On this basis, we further add two contrastive training schemes to improve the encoder.

3.2 Style Contrastive Loss

Contrastive learning is widely used in self-supervised representation learning, which is orthogonal to the training method of style transfer. We adopt the contrastive learning method to make the images with the same visual style have similar representations. As shown in Figure 2, in our scheme, the stylized result is pulled closer to the same visual style "positive" examples and pushed far away from the "negative" examples in the refined encoder's latent space. If only the target style image served as the positive example to be pulled, some content in the result would be lost. This is because the learned style embeddings are not well decoupled from the content. The content of the result is also pulled closer to the style image. To solve this problem, we obtain more positive examples through data augmentation. We use a spatial rearrangement method, the style image is divided into $n * n$ blocks, which are then randomly disrupted and recombined to obtain the images $\{I_r^0, I_r^1, \ldots, I_r^N\}$. We assume that these recombined images share the same style but different content as the original image. This method can easily construct many suitable positive examples compared with other augmentation methods. See the appendix for relevant analysis.

The features of these positive images and the stylized result from the refined encoder will be mapped into embeddings through a mapping network. We take the content images (including the original

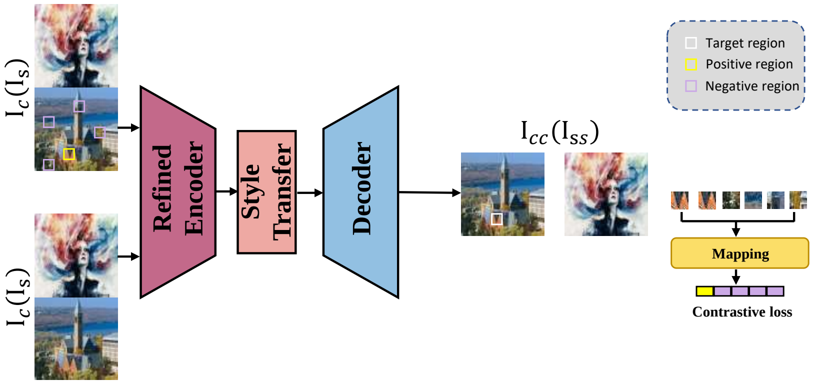


Figure 3: Our identity preserve content contrastive loss. I_c (I_s) is the input content (style) image and $I_{cc}(I_{ss})$ is the output image synthesized from this image pair (content or style). We sample some blocks of the restored and original images and compare them in a latent space. The features of the same position are pulled together, and different positions are pushed away. In this way, the encoder can retain more available details when extracting features.

content image) loaded into the batch as negative examples. Then we can formulate our style contrastive loss as follows:

$$\mathcal{L}_{s-c} = \sum_i -\log\left(\frac{\exp\left(s_o \cdot s_r^i/\tau\right)}{\exp\left(s_o \cdot s_r^i/\tau\right) + \Sigma_j \exp\left(s_o \cdot s_c^j/\tau\right)}\right), \tag{2}$$

where $s = f_s\left(E\left(I\right)\right)$, in which f_s is the style mapping network and E is our refined encoder. s_o, s_r and s_c represent style embeddings of the output result I_o, reshuffled style image I_s and content image I_c, respectively. τ is a temperature hyper-parameter to control push and pull force.

3.3 Identity Preserve Content Contrastive Loss

Content leak in the stylized results has been noticed in previous workPark & Lee (2019); An et al. (2021), which remains a problem to be solved. Some existing methods try to use an identity loss Chen et al. (2021) to preserve the content structure, but they can only train the decoder to retain the content at most, while ignoring the content leak in the encoder part. In addition, their identity losses only introduce a reconstruction loss or perceptual loss between the output image synthesized from two same content (or style) images with the original image to keep the structure. The effect is not apparent because both kinds of loss do not emphasize the local region where the content is more likely to lose.

In order to alleviate this problem, we design the identity loss as a local-wise contrastive loss similar to Park et al. (2020), which enables the encoder to retain more details when extracting features. Same as SANet, we use content (or style) features to stylize itself and obtain a restored content image. Then, as shown in Figure 3, we randomly select several blocks from the same positions of the restored image and the content image. Then, these image blocks will be mapped into latent codes that encode local structures through the same mapping network. A contrastive loss is introduced to make the latent from the same position pulled together and pushed away from other positions. Such a local-wise identity preserve content contrastive loss is expressed as:

$$\mathcal{L}_{c-c} = \sum_i -\log\left(\frac{\exp\left(\nu_{cc}^i \cdot \nu_c^i/\tau\right)}{\exp\left(\nu_{cc}^i \cdot \nu_c^i/\tau\right) + \sum_{j\neq i} \exp\left(\nu_{cc}^i \cdot \nu_c^j/\tau\right)}\right), \tag{3}$$

where $\nu^i = f_c\left(\phi_x\left(p^i\right)\right)$, in which f_c is a content mapping network, ϕ_x denotes a `ReLU_X_1` layer in our refined encoder and p^i denotes a random block of the image. ν_{cc} and ν_c represent local

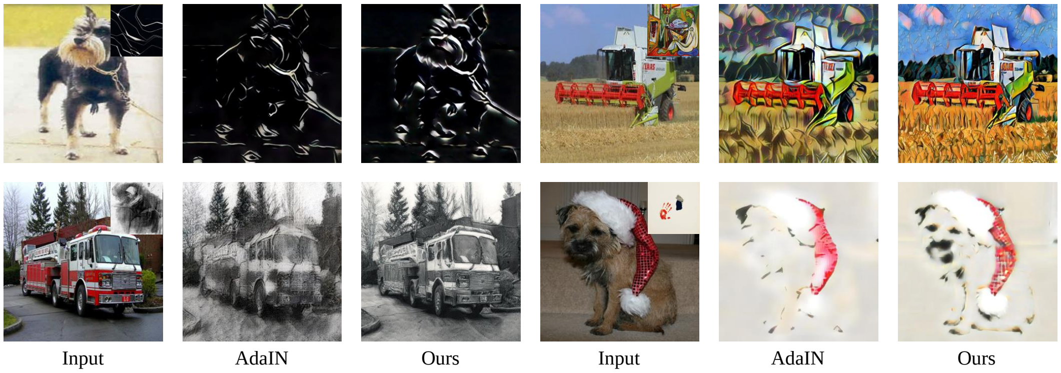


Figure 4: Experiment on AdaIN

content latent code of the image synthesized from two same content images and original content image, respectively.

3.4 Objective Function

Our training scheme can be directly added to existing encoder-decoder-based style transfer methods without changing the original architectures, and significantly improve their effectiveness. The pre-trained encoder used in the original method can be fine-tuned by optimizing the following function:

$$\mathcal{L}_{total} = \underbrace{\mathcal{L}_{ast}}_{\text{original loss}} + \underbrace{\lambda_d \sum_{i=1}^{k} \|\mathcal{F}_n^{(i)} - \mathcal{F}_o^{(i)}\|_2}_{\text{feature regularizer}} + \underbrace{\lambda_{s-c}\mathcal{L}_{c-c} + \lambda_{s-c}\mathcal{L}_{s-c}}_{\text{feature improvement}} \tag{4}$$

where L_{ast} is the loss function of the original method, $\mathcal{F}_n$ is the feature extracted by the refined encoder, and $\mathcal{F}_o$ is that of the pre-trained one. λ_d, λ_{s-s} and λ_{s-c} are the corresponding loss weights.

4 Experimental Results

In this section, we first introduce some implementation details, then add our training scheme to several existing methods and show the improvement on their methods. We also make further comparisons between our method and several baseline models. Finally, we explored the effect of our contrastive training scheme through ablation studies, especially the number of positive and negative examples.

4.1 Implementation Details

The datasets for our experiments are the commonly used MS-COCO Lin et al. (2014) (for the content images) and WikiArt Lin et al. (2014) (for the style images). Both datasets contain roughly $80,000$ training images. The optimizer (usually Adam Kingma & Ba (2014)) and the learning rate are the same as the corresponding methods. In style contrastive loss, the number of style positive examples is set to 8, and the number of negative examples is the same as batch size. The style mapping network f_s consists of a convolution layer and several subsequent MLP (multi-layer perceptron) layers, of which the last MLP layer has 128 units. The content mapping network f_c is a two-layer MLP. The number of units in the first layer is the same as the corresponding feature channel, and the second layer has 256 units. The hyper-parameter τ is set to 0.07 in both contrastive losses. The image is also processed as the previous methods: the smaller dimension of the two images is rescaled to 512 while retaining the aspect ratio and then randomly cropped to the size of $256 * 256$ pixels. Our new refined encoder is still VGG-style. For our layer split scheme, we only retain `Conv_X_1` layers and the corresponding `pooling` and `ReLU` layers of the original VGG-19.

4.2 Experiments on Existing Methods

Experiment on AdaIN. The AdaIN Huang & Belongie (2017) method presents an efficient solution for universal style transfer. It receives a content input x and a style input y, and simply aligns the

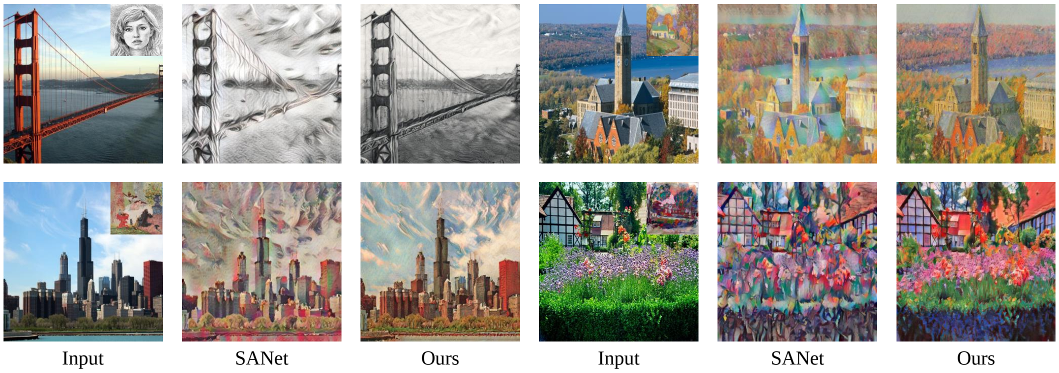


Figure 5: Experiment on SANet.

Table 1: The user study scores for different methods. The higher the better.

Stylization scheme	AdaIN	Our AdaIN	SANet	Our SANet
Preference Score	0.153	0.228	0.242	**0.377**

channel wise mean and variance of content feature maps to those of style feature maps as:

$$\mathrm{AdaIN}(x, y) = \sigma(y) \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \mu(y) \tag{5}$$

The features used here are the ReLU_4_1 layer features extracted by a pre-trained VGG-19 encoder. After the style swap operation in the feature space, the output of AdaIN can be inverted to the image space with a feed-forward decoder to get the final output. On their basis, our new encoder only retains several convolution layers from the input layer to ReLU_4_1. In the training process, the regular term is only added between the activation values of ReLU_4_1 layers of the two networks. In content contrastive loss, the ϕ we used are ReLU_1_1, ReLU_2_1, ReLU_3_1 and ReLU_4_1. The comparison results are shown in Figure 4. The results of original AdaIN retain the original color or get color that has not appeared and lost a lot of content. In contrast, our results retain more content and are more style-consistent.

Experiment on SANet. SANet Park & Lee (2019) also follows the encoder-decoder architecture, where the transfer part consists of two style-attention networks. After encoding the content and style images by the pre-trained VGG encoder, the SANet maps features from ReLU_4_1 and ReLU_5_1 features. The regular term is added on the ReLU_4_1 and ReLU_5_1 layers. The content contrastive loss is the same as AdaIN above. As shown in Figure 5, SANet is easy to migrate the patch with semantic information in the style image, resulting in strange results. In contrast, our method can ensure the correct content structure.

4.3 Quantitative Comparisons.

User Study. User study investigates users' preferences for results of different methods for more objective comparison, which is the most widely used evaluation metric in style transfer. We generate 100 stylized images using each model. These images were presented to 50 participants in random order. Participants were asked to choose their favorite image for each content-style pair. The user

Table 2: We randomly sample 5000 style images and use different encoders to calculate their style distance with the augmented images. The average distance is as follows. Our encoder has better style consistency.

Encoders	Pre-trained	Our AdaIN	Our SANet
Style Distance	37.43	7.21	**6.85**

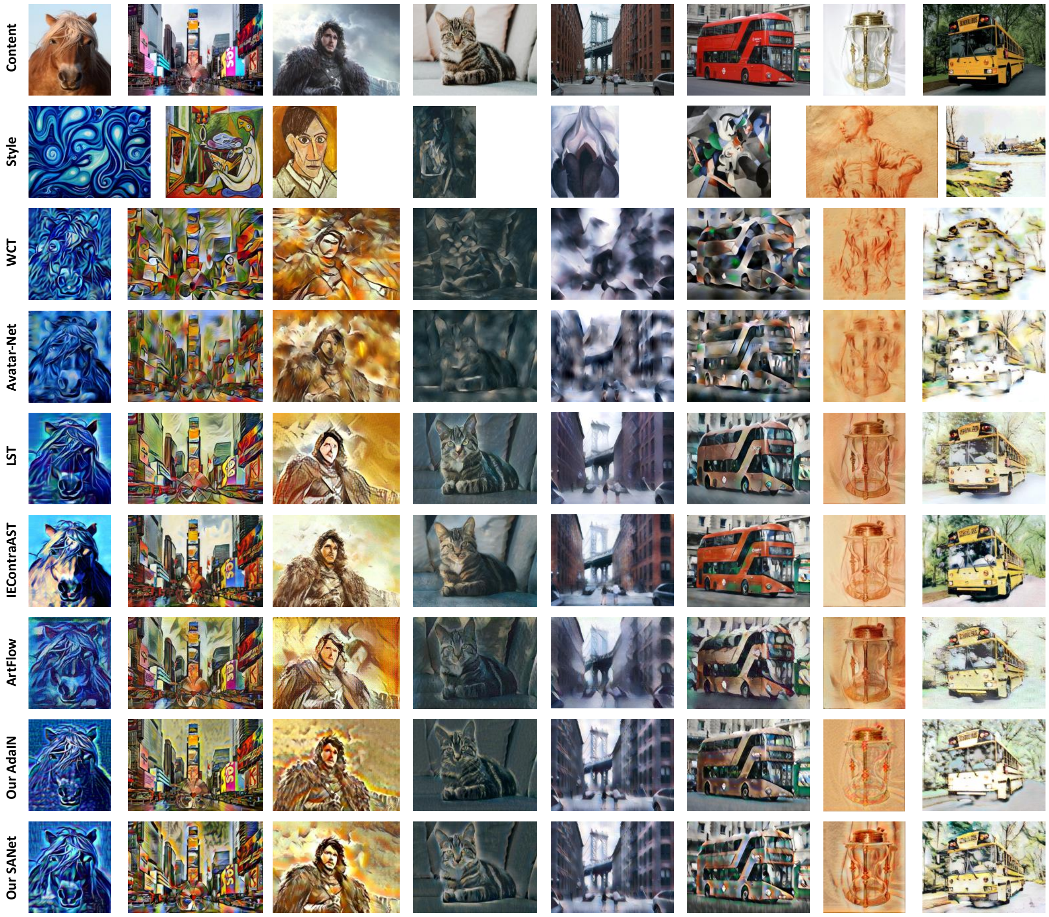


Figure 6: Comparisons with other style transfer methods.

study results are shown in Table 1. Our method's stylized images are more preferred than those of the original methods.

Style Distance. As described above, we reveal a problem of style inconsistency in the pre-trained encoder. In order to show that our encoder has better style consistency than the original one, we conduct a comparison of style distance. We use artistic images and their random augmented results as the same style images. Then, different encoders are used to calculate the style distance between the two images by Formula 1. The smaller the style distance, the better the style consistency. The comparison results are shown in Table 2. See the appendix for more relevant analysis.

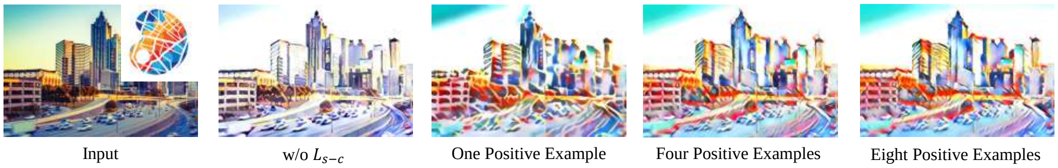
Table 3: The average LPIPS distances for different methods. The lower the better.

Inputs	WCT	Avatar	LST	AdaIN	SANet	Artflow	IECAST	Our AdaIN	Our SANet
0.214	0.452	0.347	0.326	0.353	0.334	0.387	0.337	0.312	**0.298**

LPIPS. We also extend our method to video style transfer and use LPIPS as a quantitative indicator to measure the stability and consistency of frames like the practice in Chen et al. (2021). Here, a lower LPIPS value represents better stability. Table 3 shows that our method gets the best score among all methods. The visual results of video stylization are shown in the appendix.

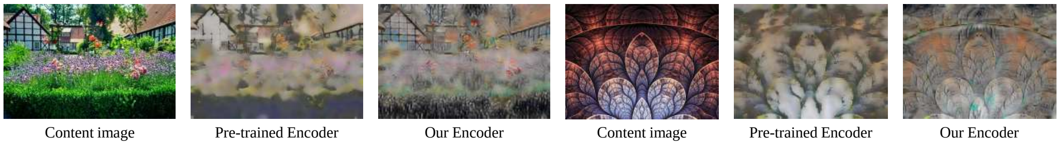
4.4 Qualitative Comparison.

We also compare our results with existing methods, including, WCT Li et al. (2017), Avatar-Net Sheng et al. (2018), LST Li et al. (2019), ArtflowAn et al. (2021) and IEContraASTChen et al.



Input w/o L_{s-c} One Positive Example Four Positive Examples Eight Positive Examples

Figure 7: Abalation study on number of style positive examples used in style contrastive learning.



Content image Pre-trained Encoder Our Encoder Content image Pre-trained Encoder Our Encoder

Figure 8: Content visualization.

(2021). Results of these methods are obtained by using the public codes and default configurations and are shown in Figure 6. The learning-free methods Li et al. (2017); Sheng et al. (2018) cannot separate style and content well, so they often fail to preserve the content structure and get distorted stylized images. Because IEcontroASTChen et al. (2021) relies on external learning, the stylized result will deviate from the style of the style image or even no style transferred at all. The flow model used in ArtFlow Chen et al. (2021) can not guarantee that the content structure is not distorted. Our method makes the stylized results achieve a better trade-off between style and content, retaining more details of original image content while keeping the style consistent with the style image.

4.5 Ablation Studies

We first study the impact of the number of examples on the results in style contrastive learning. As shown in Figure 7, when there is only one positive example, the style embedding is not well decoupled from the content, resulting in distortion in some areas. With the increasing number of positive examples, this problem will be alleviated and basically solved when set to 8. When the $Ls - c$ is removed, the style of the result is inconsistent with the style image. Although more negative examples usually lead to better performance in contrastive learning, the most crucial negative example in the style transfer task is the original content image (to distinguish from the original image style). Adding more negative examples on the basis of the original content image does not make much sense. Further, in order to prove that our training method can well preserve the content in the features extracted by the encoder, we made a visualization for the content feature. As shown in Figure 8, it can be seen that some very fine details are not lost in our encoder.

5 Discussion

Limitations. Because our training scheme needs to be added to the existing learning-based style transfer methods(such as AdaIN, SANet, etc.), and the decoder in the architecture is necessary to convert the transfer results back to the image space for content contrastive loss, our method can not be applied to some optimization-based methods.

Conclusions. In this work, we study the irrationality of the pre-trained VGG encoder used in existing style transfer methods, which shows that the style distance of some images with different styles in the feature space of the pre-trained encoder is less than that of images with the same style. This means that the encoder used in the previous method to provide supervision signals can not give a visually consistent style representation, resulting in failed style transfer result in some cases. Two contrastive training schemes are proposed to improve the encoder's latent space of the existing work, make it more style-consistent and retain more available details. Extensive experiments show the effectiveness and superiority of our method. Furthermore, this study shows that the pre-trained encoder can be replaced with a more appropriate choice in the later style transfer research.

REFERENCES

Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *CVPR*, 2021.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *NeurIPS*, 2014.

Haibo Chen, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Xing Wei, Dongming Lu, et al. Artistic style transfer with internal-external learning and contrastive learning. *NeurIPS*, 2021.

Len Du. How much deep learning does neural style transfer really need? an ablation study. In *WACV*, 2020.

Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

Bruce Gooch and Amy Gooch. *Non-photorealistic rendering*. A K Peters, 2001. ISBN 978-1-56881-133-8.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David Salesin. Image analogies. In *SIGGRAPH*, 2001.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *CVPR*, 2019.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *NeurIPS*, 2017.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Rui Liu, Yixiao Ge, Ching Lam Choi, Xiaogang Wang, and Hongsheng Li. Divco: Diverse conditional image synthesis via contrastive generative adversarial network. In *CVPR*, 2021.

Dae Young Park and Kwang Hee Lee. Arbitrary style transfer with style-attentional networks. In *CVPR*, 2019.

Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *ECCV*, 2020.

Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *ICCV*, 2019.

Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *CVPR*, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Thomas Strothotte and Stefan Schlechtweg. *Non-photorealistic computer graphics: modeling, rendering, and animation*. Morgan Kaufmann, 2002.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, 2018.

Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, and Ming-Hsuan Yang. Collaborative distillation for ultra-resolution universal style transfer. In *CVPR*, 2020.

Pei Wang, Yijun Li, and Nuno Vasconcelos. Rethinking and improving the robustness of image style transfer. In *CVPR*, 2021a.

Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. Contrastive learning based hybrid networks for long-tailed image classification. In *CVPR*, 2021b.

Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *ICCV*, 2021c.

Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Contrastive learning for compact single image dehazing. In *CVPR*, 2021.

Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, 2021.

Lu Yu, Vacit Oguz Yazici, Xialei Liu, Joost van de Weijer, Yongmei Cheng, and Arnau Ramisa. Learning metrics from teachers: Compact networks for image embedding. In *CVPR*, 2019.

Ning Yu, Guilin Liu, Aysegul Dundar, Andrew Tao, Bryan Catanzaro, Larry S Davis, and Mario Fritz. Dual contrastive loss and attention for gans. In *ICCV*, 2021.

A Appendix

A.1 Style Distance Analysis

More analysis of style distance comparisons was shown here. We randomly sampled dozens of images with different visual styles and visualized the style distance between them in Figure 9. In the original encoder, the style distance between images of different styles can be very small. We also show a quantitative comparison in Figure 10. Compared with the original pre-trained encoder, our new encoder can get a more visual consistent style representation. The distance between images of different styles is large, and the distance between images of similar visual style is small.

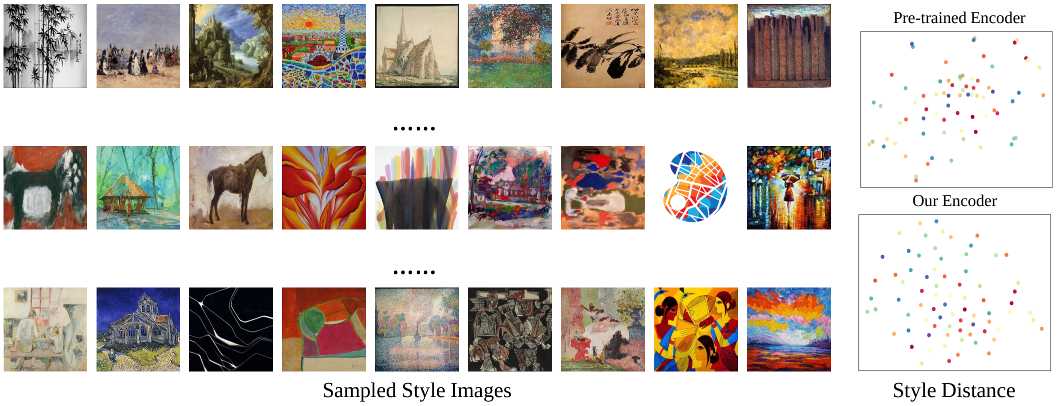


Figure 9: Visualization of style distance. We visualized the Gram matrix of each image using t-SNE.

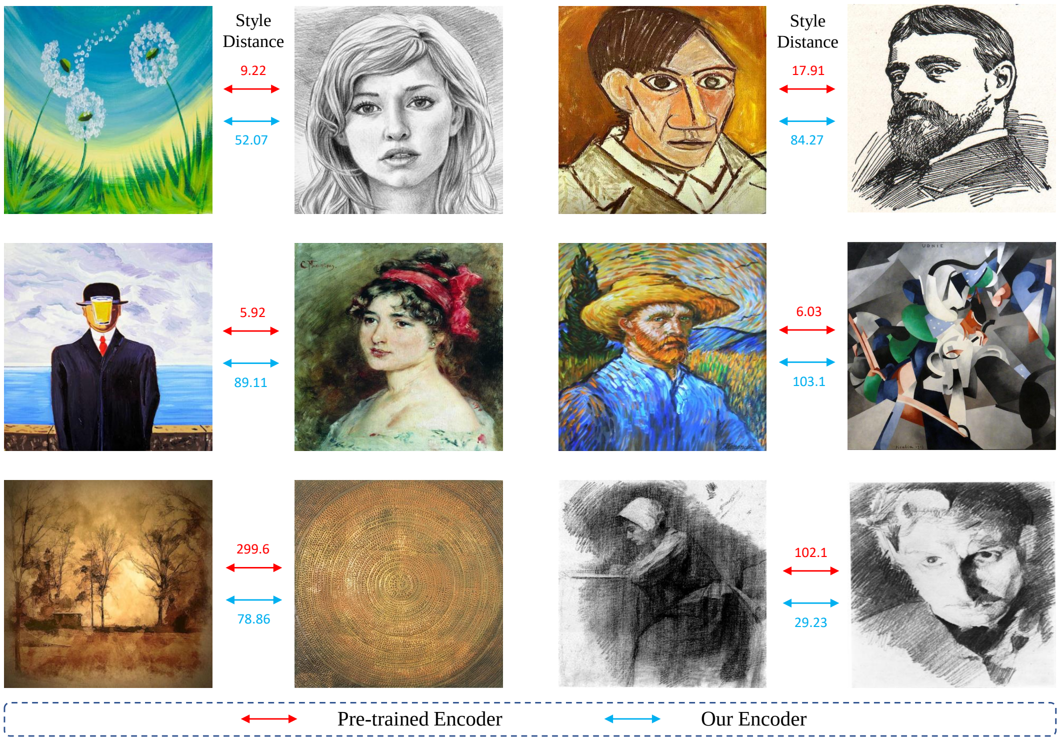


Figure 10: Visual comparison of style distance.

A.2 More Visual Results

As shown in Figure 11, we present more ablation study results to demonstrate the effectiveness of the proposed style contrastive training scheme. We also extend our method to video style, and the results are shown in Figure 12. Our method not only ensures the effect of style migration, but also ensures the consistency between video frames.

A.3 Analysis of Image Augmentation Methods

In order to make the network learn content independent style representation, we need to obtain images with the same style but different content through data augmentation. First, we cannot use methods that will change the style, such as adding noise or changing color. Second, some other spatial data augmentation methods have their own shortcomings. We show the images obtained by these methods in Figure 13. It is not easy to create many images with different contents by flip. Distortion is also difficult to disrupt the spatial layout of the original image style. Crop operation will make the image lose some critical style information. Affine transformation and rotation will make some pixels overflow the boundary and bring many blank areas. Our method of spatial rearrangement is easy to create a large number of images with the same style and different content without losing pixels.

Then we use the images obtained by these different methods to do experiments on AdaIN and show the comparison of the results in Figure 14. The network trained with images obtained by rotation and affine transformation is easy to lose content, which is caused by the blank of augmented images. The networks trained with images obtained by the other three methods tend to keep the style of the content image, because the augmented image content changes less and the network does not learn content independent style representation well.

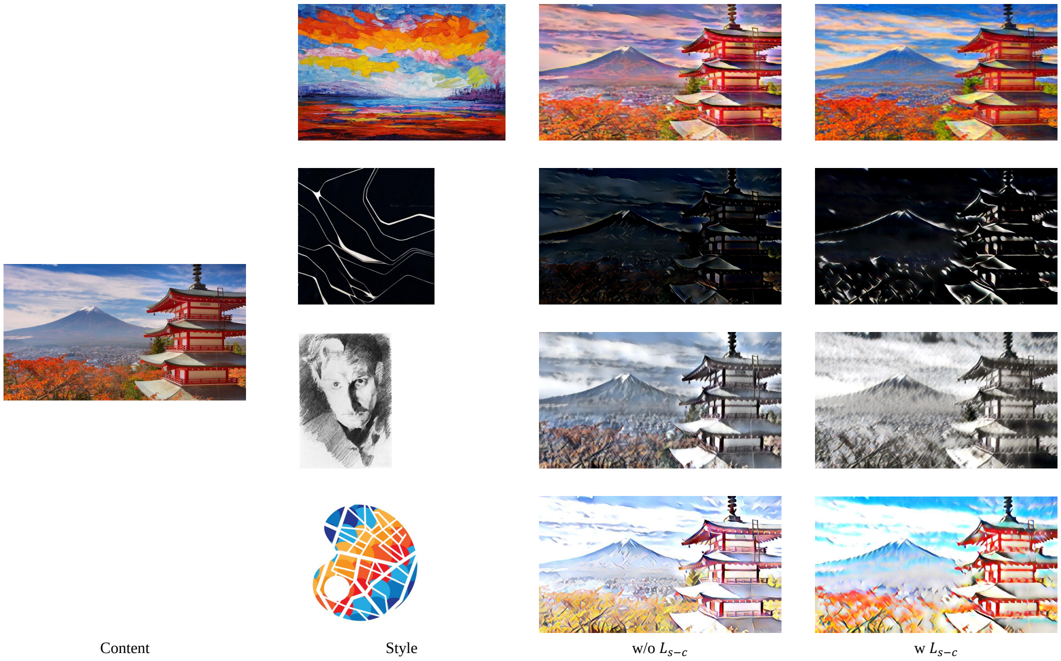


Figure 11: Ablation study of the L_{s-c}.

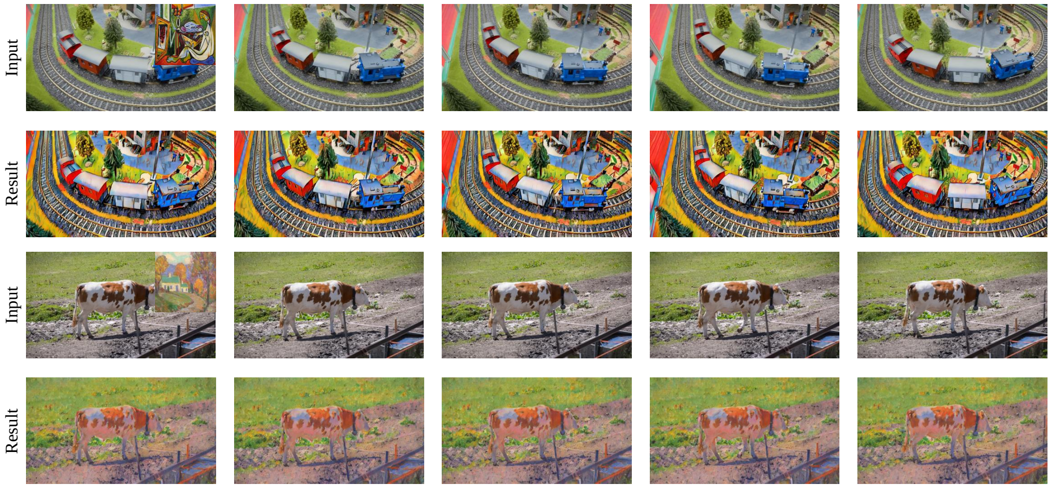


Figure 12: Video style transfer results of our SANet.

A.4 Details of User study

In this work, we conducted a user study to verify the effectiveness of our method. Here, we provide the corresponding screenshot in Figure 15. The questionnaire was distributed to all participants, and each participant was asked to complete all choices within 20 minutes.

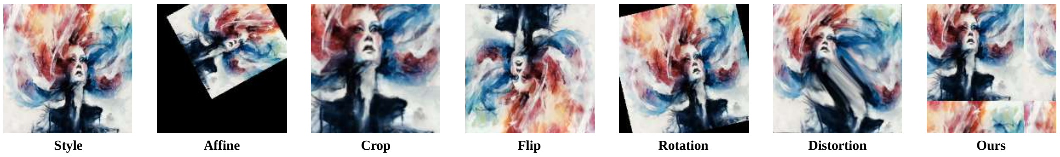


Figure 13: Comparison with other geometric data augmentation methods.

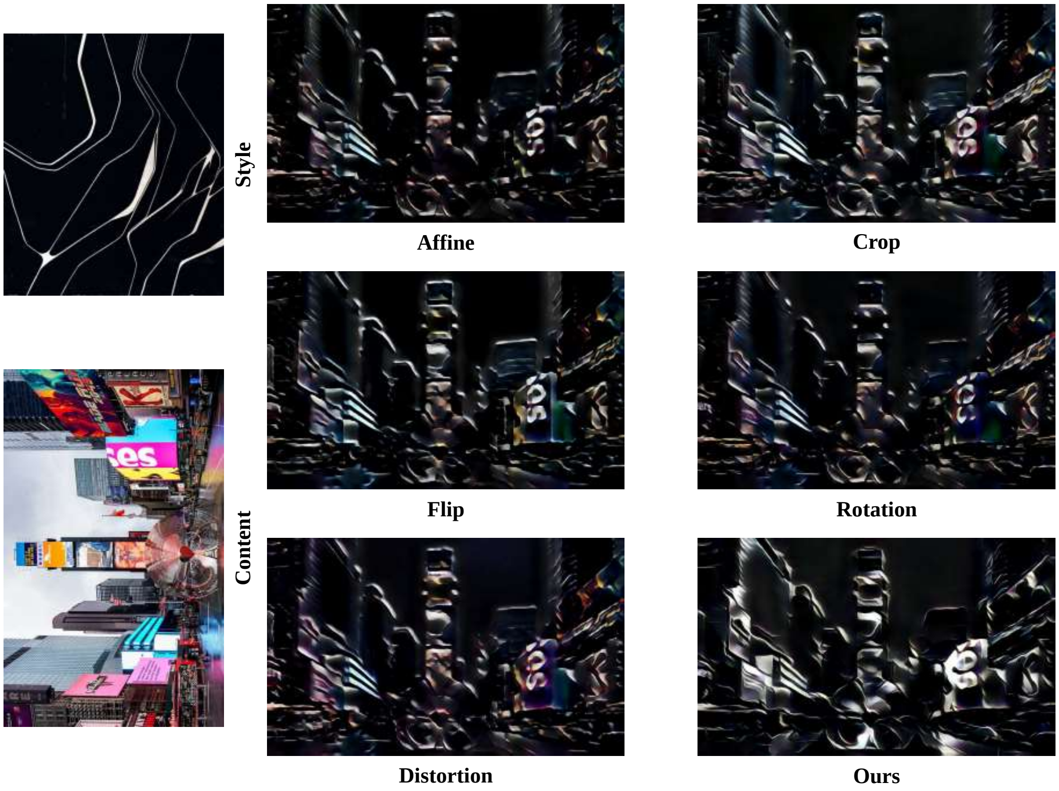
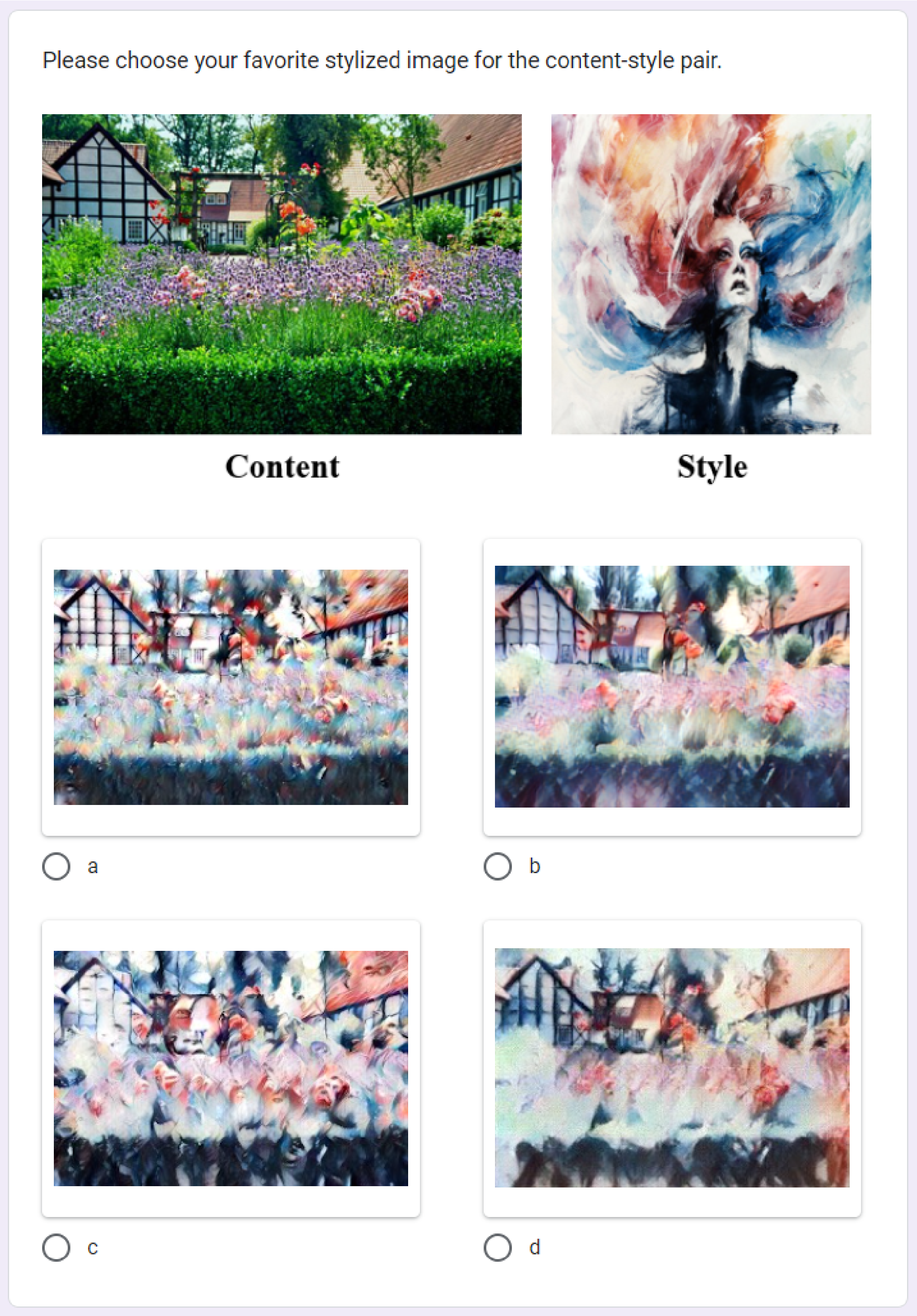


Figure 14: Comparison of results obtained by networks trained with different images.



Please choose your favorite stylized image for the content-style pair.

Figure 15: The screenshot of user study.