McCSTN: Multi-Scale Contrast and Fine-Grained Feature Fusion Networks for Subject-driven Style Transfer

Honggang Zhao, Chunling Xiao, Guozhu Jin, Jiayi Yang, Mingyong Li
College of Computer and Information Science, Chongqing Normal University
Chongqing, China
{2021210516098, 2021110516055, 2021110516041, 2023210516088}@stu.cqnu.edu.cn
Corresponding author:limingyong@cqnu.edu.cn

Abstract

Stylistic transformation of artistic images is an important part of the current image processing field. In order to access the aesthetic artistic expression of style images, recent research has applied attention mechanisms to the field of style transfer. This approach transforms style images into tokens by calculating attention and then migrating the artistic style of the image through a decoder. Due to the very low semantic similarity between the original image and the style image, this results in many fine-grained style features being discarded. This can lead to discordant artifacts or obvious artifacts. To address this problem, we propose MccSTN, a novel style representation and transfer framework that can be adapted to existing arbitrary image style transfers. Specifically, we first introduce a feature fusion module (Mccformer) to fuse aesthetic features in style images with fine-grained features in content images. Feature maps are obtained through Mccformer. The feature map is then fed into the decoder to get the image we want. In order to lighten the model and train it quickly, we consider the relationship between specific styles and the overall style distribution. We introduce a multi-scale augmented contrast module that learns style representations from a large number of image pairs. Code will be posted on https://github.com/haizhu12/MccSTN

Keywords: Attention, Aesthetics, Comparative Learning, Stylistic Transformation, Feature Fusion

1. introduction

An image is a story, and an art-style image is worth a thousand words. The purpose of image style transfer is to present the content of the source image using the characteristic elements of the style image. For example, embedding pop art stylistic features such as textures, patterns, and colors into real, everyday photographs. The main elements of our work are shown in Figure (1). Since the pioneering work of (Gatys et al., 2016), style transfers have attracted a great deal of interest from both academia and industry. The field has grown considerably due to the large influx of researchers in recent years. The main areas include improving training efficiency (Ulyanov et al., 2016), generation quality (Lin et al., 2021), generalisation ability (Hong et al., 2023), diversity of generated images (Wang et al., 2020) and user control (Kwon and Ye, 2022, 2023) . Text-driven style transfer (Zhao et al., 2023).

In order to construct a reasonable representation of style features, high-dimensional distributions of style features need to be explored to capture fine-grained features. There are several dominant approaches to the representation of style features. Examples include neural flow models (An et al., 2021), and visual transformers (Deng et al., 2022). Recent advances in image style transfer incorporate attentional mechanisms (Hong et al., 2023). The attention mechanism learns the semantic similarity relationship between patches from style images

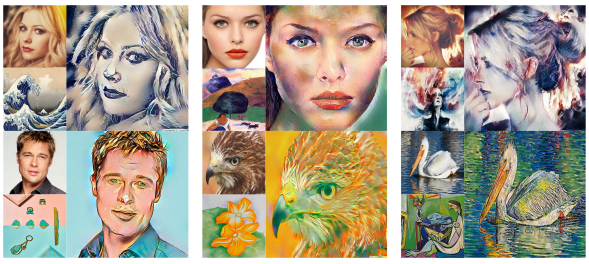

Figure 1: The generated result of our method. Artistic images provide stylistic information.

and patches from content images. This shows strong performance in guiding detail expression and retaining content information. However, they generally directly compute their attention by feeding content information and style information directly into the feature extraction module. This leads to a lot of invalid computations, increasing the amount of computation without significantly increasing the quality of the image. As shown in Figure (2), the decoded image is prone to artifacts and halos.

To solve this problem, we propose a style feature extraction module, which we call Mccformer. It fuses style features and content features and inputs the raw features of the style image and content image into the module while computing the attention score. In this way, the model can extract as many fine-grained features as possible to avoid feature loss. And it can also avoid too many invalid

LREC-COLING 2024, pages 11090–11100

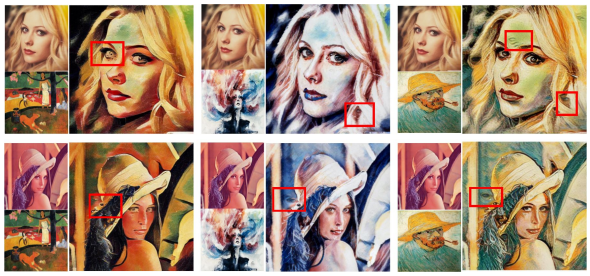

Figure 2: Problems with previous studies. Artifacts appear in line 1. An incongruous image appears in row 2. Zooming in shows it more clearly.

calculations. And the whole model is optimized by the style enhancement contrast learning module. This module for proper artistic style representation can effectively eliminate artifactual features. At the same time, Introducing a τ value makes the model more tolerant to samples with the same features. Our generated results are closer to real paintings and have better feature representation performance. **Our main contributions are as follows:**

- A feature fusion extraction module called Mccformer is proposed that fuses fine-grained features of content and style to avoid feature loss and artifacts.
- A parameter optimization method called style-enhanced contrast is proposed to improve training efficiency and image quality.
- Introduction of τ values to build flexible and effective contrast learning modules to train the correct artistic style of characteristic representation.
- Extensive experiments have demonstrated the competitiveness of our proposed method.

2. Methods

2.1. Overview of The Methodology

The dataset's content images are denoted by P_c and style images by p_s. Our aim is to train a style transformation model capable of converting any given set of content images into the desired artistic style images. The key insight is to extract aesthetic features from P_s for style transformation to synthesize images with artistic style without artifacts. Images that are comparable to real paintings. we propose a new style transformation framework called MccSTN.

As indicated in Figure (3), our MccSTN consists of four main components. 1. a pre-trained VGG encoder that projects the image into a multilevel feature embedding. 2. a feature processing module, Mccformer, that inputs the fused feature map F_m. 3. a decoder, D, that recovers the feature embedding into a stylized image. 4. a multiscale comparison module that guides the model to train the stylized migration result free of artifacts. The overall process is as follows.

1. Firstly, the content image $I_c \in P_c$ and style image $I_s \in P_s$ are input to the pre-trained VGG (Chen et al., 2021) network. The input to the VGG network is preceded by advanced downsampling. After encoding in VGG network get feature embedding token.

2. The features of the content image I_c and the features of the style image I_s are input into Mccformer to obtain the feature map $F_m = Mccformer(F_c, F_s)$.

3. The feature map F_m is input to the decoder D to obtain the style transfer image I_{cs}.

4. Input I_{cs} into the pre-trained encoder VGG to generate feature token (sharing parameters with VGG in the first step). Sample two feature tokens that match the high-dimensional feature distribution of I_{cs}, and then input it to the feature mapping module FPN to get the mapping vectors Z and Z^*, and finally sum Z and Z^* to get the mean value to get $Z+$.

5. Sample an image of another style in the dataset. Style embed this image (as in step 4) to get $Z-$.

6. Compare I_{cs} to $Z-$,$Z+$ to calculate the loss. In simple words our goal is to get I_{cs} as close to $Z+$ and as far away from $Z-$ as possible.

2.2. Mccformer Module

We propose the Mccformer module, which can adaptively fuse style features into content features by taking into account global style and local structure features through the attention mechanism. This allows the combination of VGG content and style features. According to Figure (4), Mccformer first does a fusion of vgg encoded content features and style features to get F_{cs}, then inputs the fused feature F_{cs} into the attention module to calculate the attention score, and finally encodes the target feature map F_m.

Inspired by (Wang et al., 2022; Deng et al., 2020), the inner product of channels between vectorized features can represent global features well, and channel attention can effectively improve feature representation. Specifically, given style feature $F_s \in \mathbb{R}^{C\times H\times W}$ and content feature $F_c \in \mathbb{R}^{C\times H\times W}$ from VGG (where H and W are height and width, and C is the number of channels). After obtaining the style features F_s and content features F_c, we perform regularization and convolution operations to obtain $\bar{F}_c$ and $\bar{F}_s$. The implementation is

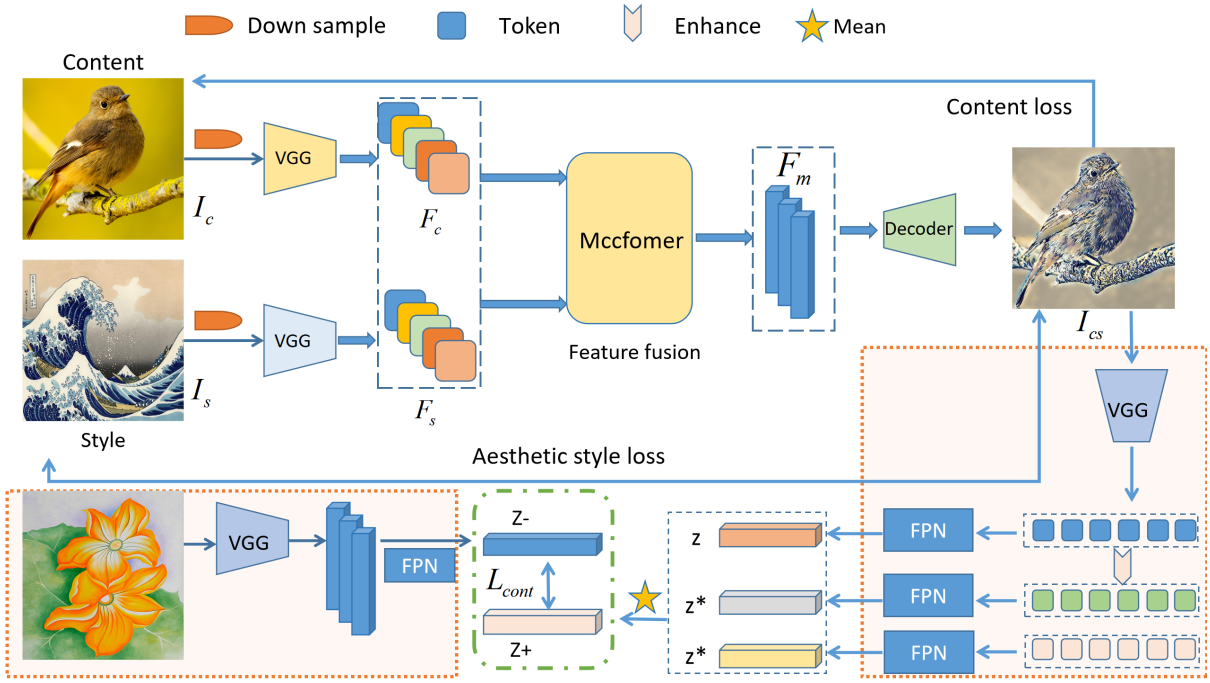

Figure 3: An overview of the methods we propose. It mainly contains the Mccfomer module for image sequence processing and the contrast learning module for model training optimization.

as follows:

$$\overline{F}_c = \Pi\left(f_{conv}\left(\mathrm{Norm}\left(F_c\right)\right)\right) \in \mathbb{R}^{C\times H\times W} \quad (1)$$

where Norm denotes regularization (mean normalization) to avoid excessive fluctuations. Where f_{conv} denotes learnable convolution. Π is a vector operation.

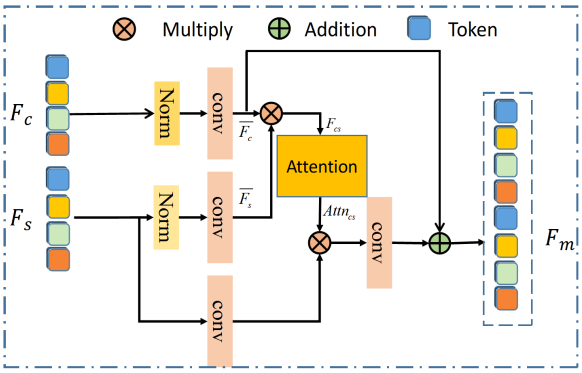

Figure 4: An overview of our Mccformer modules. The content features and style features are fed into the Mccformer module, respectively, and then the attention score is computed, and then the original feature information is added to get the final output sequence.

$\bar{F}_s$ and $\bar{F}_c$ are then dot-multiplied to obtain F_{cs}. Specifically, we input F_{cs} into the attention module to compute the attention scores between them. The output is then multiplied by the convolutional features of the style features F_s . Finally, the output is summed with the content feature F_c to obtain the final output feature map F_m. The method of realization is as follows:

$$\mathrm{Attn}_{cs} = \mathrm{Sf}\left(\Pi\left(\mathrm{attention}\left(\bar{F}_c^T \otimes \bar{F}_s\right)\right)\right) \in R^Z \quad (2)$$

where Π denotes the vector operation, attention denotes the computation of attention, sf denotes the softmax function, and T denotes the transpose operation of the vector. Z denotes the range of vector space $Z = H_c W_c \times H_s W_s$.

The implementation details of the feature map F_m are as follows:

$$F_m = f_{\mathrm{conv}}\left(\Pi\left(\left(\mathrm{Attn}_{cs}^2 \otimes F_s^T\right) \oplus \bar{F}_c\right)\right) \in R^{C\times H\times W} \quad (3)$$

where f_{conv} denotes a learnable convolution with convolution size 1x1. $\oplus$ denotes vector addition. T denotes vector transposition.

Our Mccformer module can take into account both style features and content features and introduces the original feature information after computing attention. The model can replenish the lost information, which improves the image quality and avoids artifacts and other discordant images. It also maximizes the preservation of texture features of the content image. The style attention used in our method is closely related to (Deng et al., 2020). But there are 3 main differences. (1) Our method simplifies the computational steps and processes, which

largely reduces the training time and resource consumption. At the same time the performance of feature map generation is improved and higher quality images can be generated. (2) We use fusion attention to fully exploit the semantic information of style images and content images, and high quality images can be generated under the guidance of loss function. (3) The information we input into the Mccformer module contains only style features F_s and content features F_c. This approach makes it easier to train the model and more convenient for the user during operation.

2.3. Style Enhancement Contrast

The purpose of our proposed technique is to train a topic-driven image style transformation model that needs just content pictures and style images for training. The styled picture's fine-grained characteristics are captured by the framework and incorporated into the content image. We have to produce an artistic picture that is visually appealing and devoid of artifacts.

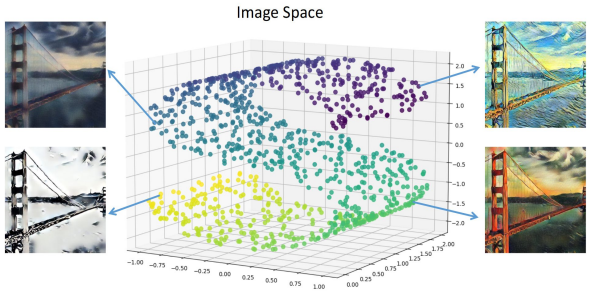

Figure 5: We use a VGG network to project the style image into high dimensional space. Visualization in 3-dimensional space can be intuitive. The enhancement process is sampling in the neighborhood of the image.

The key is to find a suitable training method. Inspired by (Zhang et al., 2023) we design the style enhancement embedding module. Specifically, we use a pre-trained VGG network to embed the image I_{cs} into the image space, which we represent using tokens, and then we use an enhancement method to increase the feature tokens by two groups (Figure (6)). Specifically, we sample two sets of tokens that match the high-dimensional distribution of the style image I_{cs} . As in Figure (5),the three sets of tokens are then projected into the latent image space to encode global style features and fine-grained style features.

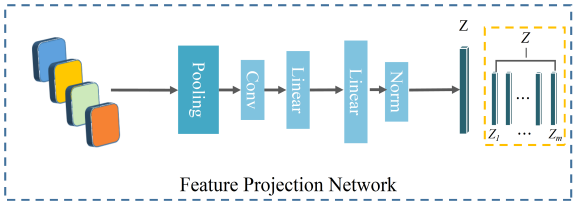

Figure 6: An overview of our feature projection network.

2.4. Comparative Training.

Due to the limited number of art-style images in the dataset and the fact that the images in the dataset have a lot of details that do not match human aesthetics, such as discordant patterns and colors, Inspired by (Zhang et al., 2023), we design style-enhanced contrast learning.

specifically, we use the VGG 19 network, which is pretrained on the WIKI dataset, for comparison training. After extracting the stylized image tokens, enhancement sampling is performed and then fed into a feature projection network (FPN) to generate style embedding codes, respectively. Finally, the style codes are summed to find a mean value. We consider this result to be a positive sample, denoted by Z^+.

At the same time, we sample other style images in the dataset (different from F_s) and project it as a style code. We consider this result as Z^- and our training goal is to minimize the distance between Z and Z^+ and maximize the distance between Z and Z- (Figure (3) bottom left). In order to prevent the training process from collapsing, these generated feature codes need to be normalized to prevent collapse (as shown in Figure (6)). Our contrast loss is defined as follows:

$$\mathcal{L}_{\text{con}} = -\sum_{i=1}^{M} \log \frac{\exp\left(D^+\right)}{\exp\left(D^+\right) + \sum_{j=1}^{N} \exp\left(D^-\right)} \tag{4}$$

where "." denotes the dot product operation of two vectors. $D^+ = \mathbf{z}_i \cdot \mathbf{z}_i^+ / \tau$. $D^- = \mathbf{z}_i \cdot \mathbf{z}_{i_j}^- / \tau$. N denotes the number of negative samples sampled.

Perceiving hardness aids in recognizing distinct characteristics that are evenly spread out but may have a limited tolerance for samples that are identical in meaning. The magnitude of the penalty imposed on challenging negative samples is determined by the temperature parameter (τ). As the temperature decreases, the penalty becomes more focused in regions with high similarity, whereas as the temperature increases, the distribution of penalties becomes more even, resulting in all negative samples being penalized equally. An association between uniformity, tolerance, and temperature is established.

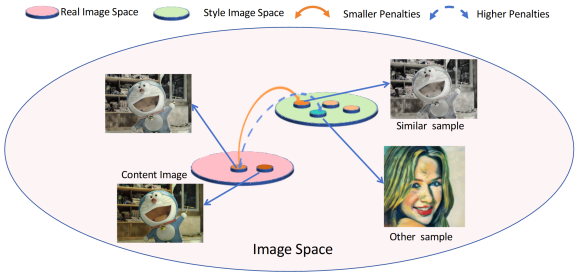

Figure 7: Similar negative samples with small τ values have small penalties. Unsimilar negative samples with larger τ values have larger penalties.

2.5. Penalty Parameter

We introduce the variable τ to regulate the dispersion of the negative gradient (Zhang et al., 2023). Decreasing the value of τ increases the emphasis on the anchor point's nearest neighbours, whilst increasing τ equally penalises negative samples. This is seen in Figure (7). When the value of τ is held constant, the magnitude of the gradient for the positive samples is equivalent to the total of the gradients for all the negative samples. Prior research on τ analysis has mostly examined the lack of uniformity in penalties for negative samples inside an anchor (Wang and Liu, 2021), or the cumulative penalties of various anchors within a training batch. Contrary to prior work, the current study specifically examines the ratio of fines between positive and negative samples.

This work considers the similarities between the style codes of other creative images I_s^- and the reference style I_s, and provides an input-dependent strategy to calculate τ. The τ rises with the number of extremely similar samples in the memory bank. To do this, τ is represented using the sigmoid function, which is a monotonic function with defined upper and lower limits. In order to align with the centre of the sigmoid function, the independent variable of image similarity needs to be normalised to a distribution with a mean of 0. The distribution of image similarity is presumed to follow a Gaussian distribution.

In order to ensure consistency in image similarity during the training process, the mean and variance are calculated. As the training progresses and more samples are included, the data distribution's mean and variance are estimated. Here are the recursive rules: The new mean is determined by first updating the average and then evaluating the average similarity of each new image in relation to the known mean similarity. The new variance is calculated by adjusting the existing variance based on the difference between the similarity of each new image and the average similarity, taking into account their respective weights. We calculate our input-dependent τ as follows:

$$\tau = \frac{1}{1 + \exp\left(-\left(\sum_{n=1}^{N} g\left(\mathrm{s_n}\right) - \mu\right) \cdot \sigma\right)} + \beta \quad (5)$$

where μ and σ denote the estimates of the mean and standard deviation of $\sum_{n=1}^{N} g\left(\mathrm{s_n}\right)$, and β denotes the lower limit of τ. β is set to 0.05.

2.6. Style Code Embedding

Feature Projection Network. (Zhang et al., 2023) We aim to create a comprehensive framework for migrating artistic images to natural photos, preserving both the local stroke characteristics and overall appearance. An essential element is to identify appropriate style representations that may be utilised to differentiate between various styles and then direct the creation of stylized images. To achieve this objective, we present the feature projection module, comprising a style feature extractor and a Feature Projection Network (FPN). In the FPN, As shown in Figure (3), instead of utilising layer-specific features or combining several layers, features from different layers are projected onto a distinct latent image space to capture both local and global style cues.

More precisely, we utilised VGG-19 (Simonyan and Zisserman, 2014), a pre-trained model on ImageNet, and made further adjustments to optimise its performance. The model amassed a total of 18,000 art images spanning over 50 distinct categories. The M-layer feature maps extracted from VGG-19 were chosen as the inputs for the multilayer projector. The mean and maximum values of the characteristics were captured using maximum pooling and average pooling techniques. The maximum pooling and average pooling layers are employed to extract the highest and mean values of the features, respectively. The FPN comprises a pooling layer, a convolutional layer, and numerous multilayer linear layers. These layers are responsible for projecting the style features into a collection of K-dimensional latent style codes $\left\{\mathrm{z}_i \mid i \in [1, M], \mathrm{z}_i \in \mathbb{R}^K\right\}$.

3. Generalized Loss Functions

For generic style transfer, as in case of (Huang and Belongie, 2017), we compute the content loss L_c and the style loss L_s, which are implemented as follows, with the goal to preserve as much of the texture features of the content image and the color, pattern, and other features of the style image as possible in the generated image:

$$\mathcal{L}_c = \|\mathrm{Norm}\left(I_{cs}\right) - \mathrm{Norm}\left(I_c\right)\|_2 \quad (6)$$

where Norm denotes regularization.

Table 1: The inference time represents the time used to process an image. The best of all results are in bold, and the second-best are underlined.

Methods	Inference time↓	Content loss↓	LPIPS↓	Deception Rate↑
MANet (Deng et al., 2020)	43ms	0.155	0.338	39.5%
ArtFlow (An et al., 2021)	118ms	<u>0.121</u>	0.314	37.6%
IEContraAST (Chen et al., 2021)	62ms	0.134	**0.305**	<u>61.4%</u>
AesUST (Wang et al., 2022)	59ms	0.143	0.334	58.5%
CAST (Zhang et al., 2023)	436ms	0.160	0.328	61.7%
MicroAST (Wang et al., 2023)	**23ms**	0.177	0.340	54.1%
MccSTN(Ours)	<u>41ms</u>	**0.109**	<u>0.310</u>	**69.2%**

We compute the style loss so that the generated image retains as many of the fine-grained features of the original style image, such as pattern, texture, and color, as possible.

$$\begin{aligned}\mathcal{L}_s = &\left\|\mu\left(I_{cs}\right)-\mu\left(I_s\right)\right\|_2 \\ &+\left\|\sigma\left(I_{cs}\right)-\sigma\left(I_s\right)\right\|_2\end{aligned} \tag{7}$$

where μ and σ are the mean and standard deviation of the channel, respectively.

Furthermore, in order to limit the loss of content features and preserve more texture and global features, we use the identity loss L_{id} to constrain the identity mapping between content features and style features.

$$\mathcal{L}_{id} = \left\|I_{cc}-I_c\right\|_2+\left\|I_{ss}-I_s\right\|_2 \tag{8}$$

Where I_{cc} denotes the use of image I_c as the result of generating content images and style images, and I_{ss} denotes the use of style image Is as the result of generating content images and style images.

The overall optimization objective is:

$$L_{\text{total}} = \lambda_1 L_{\text{cont}} + \lambda_2 L_c + \lambda_3 L_s + \lambda_4 L_{id} \tag{9}$$

where λ is a weighting factor that adjusts the weight of the loss term.

4. Experiments

4.1. Implementation Details

We follow the multilevel strategy of (Huang and Belongie, 2017) by integrating two Mccformer modules on the index 3 and index 4 layers of VGG-19, respectively. Our content dataset is MS-COCO (Lin et al., 2014) and style dataset is WikiArt (Phillips and Mackintosh, 2011). Both datasets contain about 80,000 training images. We use the Adam optimizer with a learning rate of e^{-4} and a batch size of 4. A total of 160,000 iterations were used for training. During training, all images are rescaled to 512 while preserving the aspect ratio and then randomly cropped to 256×256 pixels. We use the PyTorch deep learning framework, and all experiments are performed on NVIDIA RTX 3090 24GB GPUs.

4.2. Comparison of SOTA Methods

We compare our method to six of the state-of-the-art methods, specifically Multi-Adaptation (Deng et al., 2020), IEContraAST (Chen et al., 2021), MicroAST (Wang et al., 2023), AesUST (Wang et al., 2022), CAST (Zhang et al., 2023), and ArtFlow (An et al., 2021). All baselines use default configurations and publicly available code.

As shown in Table (1), the inference speed and other common parameters are compared. We use inference picture time to explore the degree of lightness of the modeling approach. We use content loss to indicate the extent to which the model loses content features during feature embedding. We use LPIPS to explore the quality of the final generated image. Deception loss (Kotovenko et al., 2019) is to measure whether the generated result has obvious artifacts. Specifically, it is a measure of whether the generated image differs from a real painting. We conducted a user study to calculate deception loss, i.e., the number of times a generated image was guessed to be a "real painting". We randomly selected 40 composite images for each method and asked 50 subjects to guess whether it was a real painting.

According to Table (1), our method reaches the state-of-the-art in almost all the domains, e.g., our LPIPS reaches the minimum, indicating that our method loses the least number of content features during the style migration process. The reception rate is the highest, indicating that the quality of our images is better and more preferred by the users.

4.3. Qualitative Comparison

We show the results of the qualitative comparison, as shown in Figure (8). ArtFlow, MicroAST, etc. produce obvious artifacts and discordant patterns. This is what we do not want to see. MAnet, IEContraAST, etc. can generate concise and effective results, but some of the generated results have distorted structures and artifacts. AesUST, CAST, preserves very well in terms of content features but lacks the ability to perceive certain complex

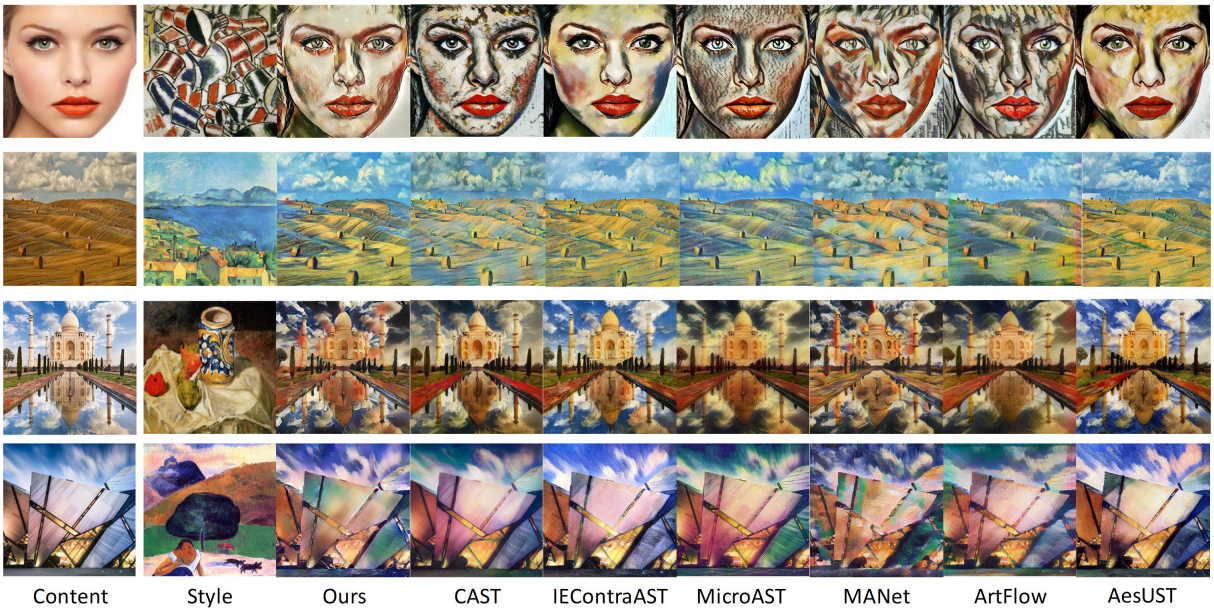

Figure 8: Qualitative comparison results. Our method generates higher quality images.

textures and patterns. Our method generates high-quality images. These problems are avoided.

4.4. User Preference Study

because of the unique qualities of creative pictures. It is not a good idea to represent the quality of image production using mathematical formulas. We created a user preference research as a result. Users' choices for photographs vary, and even the same image will receive varying ratings. We polled fifty users about their preferences. Finding more aesthetically pleasing photos that are favored by a larger number of individuals is our aim. To be more precise, we contrasted our approach with the best available technique to select photographs with greater aesthetic appeal from alternatives A and B. In order to select photos with greater visual appeal, we also contrasted our approach with the best one currently in use. We conducted individual tests with each user and noted their selections. Every time we compare our approach with one of the other state-of-the-art ways, we select one hundred styles for every content image and ask the user to select the method that produces an image that they find more aesthetically pleasing out of the two. Figure (9) shows the results of comparing our method with other state-of-the-art methods.

4.5. Degree of Stylization Control

By adjusting the various in Eq.(9), we can regulate the level of stylization applied to an image. The feature maps are altered by changing the weights of

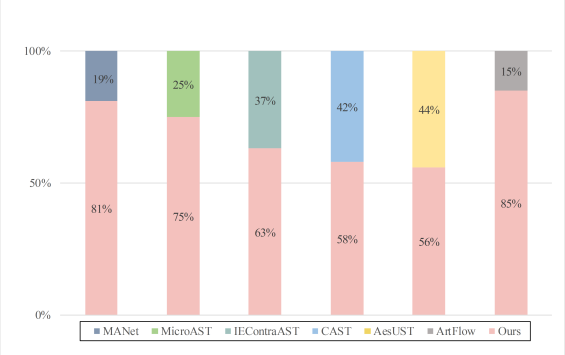

Figure 9: The results of our user preference study.

various loss functions to generate images with different degrees of stylization. This can be done with good results, but it requires retraining the model and is very user-unfriendly. To achieve runtime control, inspired by (Huang and Belongie, 2017), we introduce an interpolation method to control the degree of stylization as shown in Figure (10).

In the testing process, we first input the content image I_c and the style image Is into the model and output the feature map F_m. Then we use the content image I_c as the content input and the style input. Output feature map F_{cm}. φ value is used to control the weight value of feature maps Fm and F_{cm}, fusing the two feature maps to get F_{mcm}, the realization formula is as follows.

$$F_{mcm} = \varphi F_m + (1 - \varphi) F_{cm} \tag{10}$$

When $\varphi = 1$ is satisfied, the entire model will be

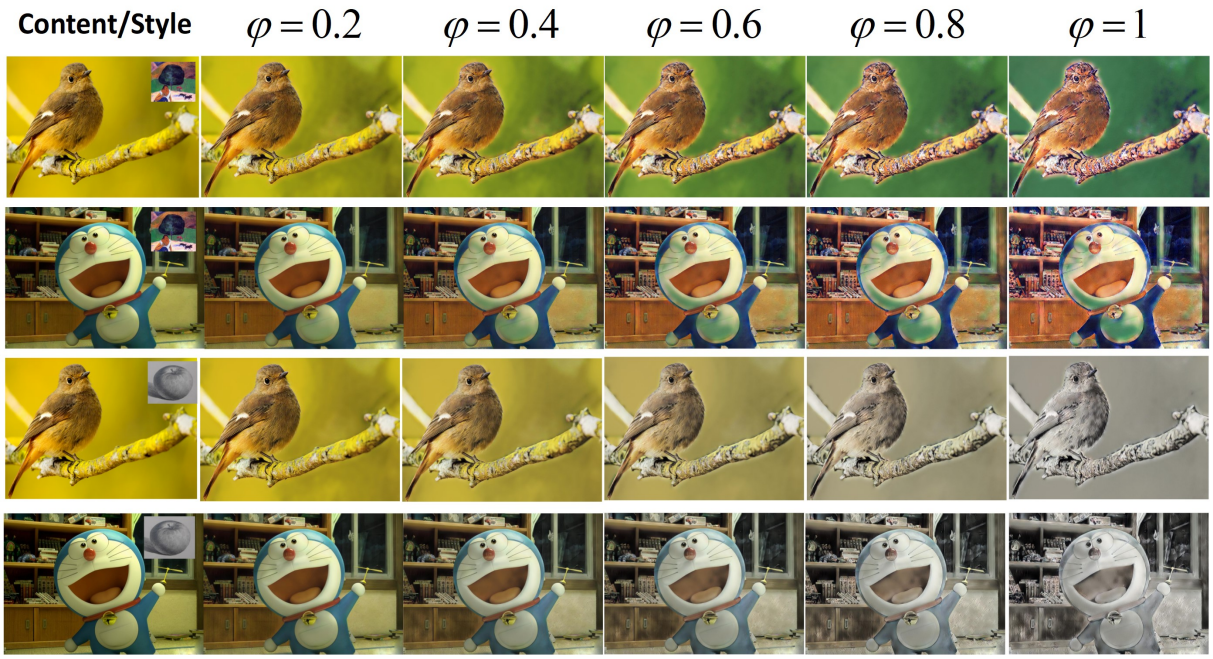

Figure 10: Example of runtime style control with a style image in the upper right corner of the content image. different ϕ values result in different degrees of stylization.

fully stylized output. When $\varphi = 0$, the entire model will be an image reconstruction.

5. Limitations

Our method easily extracts fine-grained features from style images, but for many with multiple distinctive features, Our method may give the wrong results. There are many features that we don't want that are also embedded in the feature map F_m by the Mccformer module. In short, there will be some difference between the style of the image recognized by the human eye and the style recognized by the neural network. In our later work, we will add more manual annotation, which can make the results generated by the model more suitable for human aesthetics. Our work does not compute attention scores for content images or style image sheets separately, which perhaps affects the model's ability to recognize image styles. We will introduce the multiple attention module in our future work, which we believe can significantly improve the performance of the model.

6. Conclusion

We propose a simple and effective style transfer network that can easily embed the fine-grained features of style images into content images. Specifically, we fuse content images and stylized images by computing an attention score. We call it Mccformer. This method fuses the generated feature maps to avoid artifacts. We also propose the style enhancement contrast method to optimize the parameters. Specifically, this method improves the training efficiency and solves the problem of insufficient supervised information in the dataset. Through experimental exploration, our method has achieved its current state-of-the-art optimal performance. Artifacts are almost eliminated while preserving content image texture features, and style control is achieved. No additional artistic feature embedding module is required to generate highly aesthetic images. Extensive experiments show the superiority of our image conversion method.

7. Acknowledgement

This work was partially supported by the Chongqing Natural Science Foundation of China(Grant No. CSTB2022NSCQ-MSX1417), the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJZD-K202200513) and Humanities and social science research project of Chongqing Municipal Education Commission(22SKGH100), Research on the Influencing Factors of Digital Competence of Primary and Secondary School Teachers from the Perspective of Digital Transformation of Education(Grant No. YZH23015).

8. Ethical Approval

The data for the images in this paper were obtained from public datasets to ensure that this study would not cause any mental or physical harm to the subjects and would not be detrimental to their safety and rights.

9. References

Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. 2021. Artflow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 862–871.

Haibo Chen, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, Dongming Lu, et al. 2021. Artistic style transfer with internal-external learning and contrastive learning. *Advances in Neural Information Processing Systems*, 34:26561–26573.

Yingying Deng, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, and Changsheng Xu. 2021. Arbitrary video style transfer via multi-channel correlation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1210–1217.

Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. 2022. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11326–11336.

Yingying Deng, Fan Tang, Weiming Dong, Wen Sun, Feiyue Huang, and Changsheng Xu. 2020. Arbitrary style transfer via multi-adaptation network. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2719–2727.

Jakub Fišer, Ondřej Jamriška, Michal Lukáč, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sỳkora. 2016. Stylit: illumination-guided example-based stylization of 3d renderings. *ACM Transactions on Graphics (TOG)*, 35(4):1–11.

Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13.

Leon Gatys, Alexander S Ecker, and Matthias Bethge. 2015a. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015b. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423.

Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2017. Controlling perceptual factors in neural style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3985–3993.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Kibeom Hong, Seogkyu Jeon, Junsoo Lee, Namhyuk Ahn, Kunhee Kim, Pilhyeon Lee, Daesik Kim, Youngjung Uh, and Hyeran Byun. 2023. Aespa-net: Aesthetic pattern-aware style transfer networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22758–22767.

Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510.

Yongcheng Jing, Xiao Liu, Yukang Ding, Xinchao Wang, Errui Ding, Mingli Song, and Shilei Wen. 2020. Dynamic instance normalization for arbitrary style transfer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 4369–4376.

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

Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. 2019. Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10051–10060.

Dmytro Kotovenko, Artsiom Sanakoyeu, Pingchuan Ma, Sabine Lang, and Bjorn Ommer. 2019. A content transformation block for image style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10032–10041.

Gihyun Kwon and Jong Chul Ye. 2022. Clipstyler: Image style transfer with a single text condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18062–18071.

Gihyun Kwon and Jong Chul Ye. 2023. One-shot adaptation of gan in just one clip. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2018. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51.

Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2017. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30.

Tianwei Lin, Zhuoqi Ma, Fu Li, Dongliang He, Xin Li, Errui Ding, Nannan Wang, Jie Li, and Xinbo Gao. 2021. Drafting and revision: Laplacian pyramid network for fast high-quality artistic style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5141–5150.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. 2021. Adaattn: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6649–6658.

Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. 2019. Exploiting unlabeled data in cnns by self-supervised learning to rank. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1862–1878.

P.-G. Maillot. 1990. Using quaternions for coding 3D transformations. In A. S. Glassner, editor, *Graphic Gems*, pages 498–515. Academic Press, Boston, MA.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Dae Young Park and Kwang Hee Lee. 2019. Arbitrary style transfer with style-attentional networks. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5880–5888.

Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei Efros, and Richard Zhang. 2020. Swapping autoencoder for deep image manipulation. *Advances in Neural Information Processing Systems*, 33:7198–7211.

Fred Phillips and Brandy Mackintosh. 2011. Wiki art gallery, inc.: A case for critical thinking. *Issues in Accounting Education*, 26(3):593–608.

A. Sanna and P. Montuschi. 1997. A new algorithm for the rendering of CSG scenes. *The Computer Journal*, 9:555–564.

Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. 2018. Visual permutation learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):3100–3114.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Jan Svoboda, Asha Anoosheh, Christian Osendorfer, and Jonathan Masci. 2020. Two-stage peer-regularized feature recombination for arbitrary image style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13816–13825.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. 2016. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*.

Bin Wang, Wenping Wang, Huaiping Yang, and Jiaguang Sun. 2004. Efficient example-based painting and synthesis of 2d directional texture. *IEEE Transactions on Visualization and Computer Graphics*, 10(3):266–277.

Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504.

Zhizhong Wang, Zhanjie Zhang, Lei Zhao, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. 2022. Aesust: towards aesthetic-enhanced universal style transfer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1095–1106.

Zhizhong Wang, Lei Zhao, Haibo Chen, Lihong Qiu, Qihang Mo, Sihuan Lin, Wei Xing, and Dongming Lu. 2020. Diversified arbitrary style transfer via deep feature perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7789–7798.

Zhizhong Wang, Lei Zhao, Zhiwen Zuo, Ailin Li, Haibo Chen, Wei Xing, and Dongming Lu. 2023. Microast: towards super-fast ultra-resolution arbitrary style transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2742–2750.

Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. 2021a. Contrastive learning for compact single image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10551–10560.

Xiaolei Wu, Zhihao Hu, Lu Sheng, and Dong Xu. 2021b. Styleformer: Real-time arbitrary style transfer via parametric style composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14618–14627.

Zijie Wu, Zhen Zhu, Junping Du, and Xiang Bai. 2022. Ccpl: contrastive coherence preserving loss for versatile style transfer. In *European Conference on Computer Vision*, pages 189–206. Springer.

Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang. 2019. Attention-aware multi-stroke style transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1467–1475.

Yuxin Zhang, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, Tong-Yee Lee, and Changsheng Xu. 2022. Domain enhanced arbitrary image style transfer via contrastive learning. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–8.

Yuxin Zhang, Fan Tang, Weiming Dong, Haibin Huang, Chongyang Ma, Tong-Yee Lee, and Changsheng Xu. 2023. A unified arbitrary style transfer framework via adaptive contrastive learning. *ACM Transactions on Graphics*.

Honggang Zhao, Guozhu Jin, Xiaolong Jiang, and Mingyong Li. 2023. Sde-rae: Clip-based realistic image reconstruction and editing network using stochastic differential diffusion. *Image and Vision Computing*, page 104836.