Domain Enhanced Arbitrary Image Style Transfer via Contrastive Learning

Yuxin Zhang\(^1\,2\), Fan Tang\(^3\)*, Weiming Dong\(^1,2\)*, Haibin Huang\(^4\)
Chongyang Ma\(^4\), Tong-Yee Lee\(^5\), Changsheng Xu\(^1,2\)
\(^1\)NLPR, Institute of Automation, Chinese Academy of Sciences
\(^2\)School of Artificial Intelligence, UCAS
\(^3\)School of AI, Jilin University
\(^4\)Kuaishou Technology
\(^5\)National Cheng-Kung University

Figure 1: Style transfer results using our method, which can robustly and effectively handle various painting styles. The input content image is in the top-left corner and the style reference is shown as the inset for each result. Our method can faithfully capture the style of each painting and generate a result with a unique artistic visual appearance. Artists of style images: (b) August Macke, (c) August Macke, (d) August Macke, (e) Claude Monet, (f) Vincent van Gogh, (g) Wassily Kandinsky, (h) Vincent van Gogh, (i) Hiroshige, (j) Guanzhong Wu, (k) Edouard Cortes, and (l) Howard Finster.

ABSTRACT
In this work, we tackle the challenging problem of arbitrary image style transfer using a novel style feature representation learning method. A suitable style representation, as a key component in image stylization tasks, is essential to achieve satisfactory results. Existing deep neural network based approaches achieve reasonable results with the guidance from second-order statistics such as Gram matrix of content features. However, they do not leverage sufficient style information, which results in artifacts such as local distortions and style inconsistency. To address these issues, we propose to learn style representation directly from image features instead of their second-order statistics, by analyzing the similarities and differences between multiple styles and considering the style distribution. Specifically, we present Contrastive Arbitrary Style Transfer (CAST), which is a new style representation learning and style transfer method via contrastive learning. Our framework consists of three key components, i.e., a multi-layer style projector for style code encoding, a domain enhancement module for effective learning of style distribution, and a generative network for image style transfer. We conduct qualitative and quantitative evaluations comprehensively to demonstrate that our approach achieves significantly better results compared to those obtained via state-of-the-art methods. Code and models are available at https://github.com/zyxElsa/CAST_pytorch.

CCS CONCEPTS
• Computing methodologies → Image processing.
KEYWORDS
Arbitrary style transfer, contrastive learning, style encoding

1 INTRODUCTION
If a picture is worth a thousand words, then an artwork tells the whole story. Art styles, which describe the way the artwork looks, are the manner in which the artist portrays his or her subject matter and how the artist expresses his or her vision. Style is determined by the characteristics that describe the artwork, such as the way the artist employs, forms, colors, and composition. Artistic style transfer, as an efficient way to create a new painting by combining the content of a natural image and the style of an existing painting image, is a major research topic in computer graphics and computer vision [Jing et al. 2020b; Liao et al. 2017], with style representation as the most important issue.

Since Gatys et al. [2016] proposed to use Gram matrix as artistic style representation, high-quality visual results are generated by advanced neural style transfer networks. Despite the remarkable progress made in the field of arbitrary image style transfer, the second-order feature statistics (Gram matrix or mean/variance) style representation has restricted further development and application. As shown in Figure 1, the appearances of different artwork styles vary considerably in terms of not only the colors and local textures but also the layouts and compositions. Figures 2d and 2e show the results of two recently proposed state-of-the-art style transfer approaches. We observe that aligning the distributions of neural activation between images using second-order statistics results in difficulty to capture the color distribution or the special layouts, or imitate specific detailed brush effects of different styles.

In this paper, we revisit the core problem for neural style transfer, that is, the proper artistic style representation. The widely used second-order statistics as a global style descriptor can distinguish styles to some extent, but they are not the optimal way to represent styles. By second-order statistics, arbitrary stylization formulates styles through artificially designed image features and loss functions in a heuristic manner. In other words, the network learns to fit the second-order statistics of the style image and generated image, instead of the style itself. Exploring the relationship and distribution of styles directly from artistic images instead of using pre-defined style representations is worthwhile.

We introduce contrastive learning to consider the positive and negative relationships between styles, and we use DE to learn the distribution of overall art image domains. To capture the style features at various scales, our MSP module projects the features of each layer of the style image to the corresponding style encoding space.

Our contribution can be summarized as follows:
• We propose an MSP module for style encoding and a novel CAST model for encoder-transformation-decoder-based arbitrary style transfer without using the second-order statistics as style representations.
• We introduce contrastive learning and domain enhancement by considering the relationships between positive and negative examples as well as the global distribution of styles, which solves the problem that existing style transfer models cannot fully utilize a large amount of style information.
• Experiments show that our method achieves state-of-the-art style transfer results in terms of visual quality. A challenging subjective survey was conducted, as inspired by the Turing test, to show that output of CAST could mislead participants from telling the fake painting images from real ones.

2 RELATED WORK
Image style transfer. Traditional style transfer methods such as stroke-based rendering [Fisher et al. 2016] and image filtering [Wang et al. 2004] typically use low-level hand-crafted features. Gatys et al. [2016] and the follow-up variants [Gatys et al. 2017; Kolkin et al. 2019] demonstrate that the statistical distribution of features extracted from pre-trained deep convolutional neural networks can capture style patterns effectively. Although the results are remarkable, these methods formulate the task as a complex optimization problem, which leads to high computational cost. Some recent approaches rely on a learnable neural network to match the statistical information in feature space for efficiency. Per-style-per-model methods [Gao et al. 2020; Johnson et al. 2016; Puy and Perez 2019] train a specific network for each individual style. Multiple-style-per-model methods [Chen et al. 2017; Dumoulin et al. 2017;
To consider both shallow and deep features for attention (AdaAttN) to prevent content leak during universal style transfer. Liu et al. [2021] proposed contrastive learning in image dictionary [Santa Cruz et al. 2019], geometric prediction [Liu et al. 2019], and image translation. Contrastive learning has been used in collection style transfer, which considers other artistic images in the style bank will be used as negative samples. We compute a contrastive style loss $L_{contra}$ based on these style codes. DE module is based on the adversarial loss $L_{adv}$ and the cycle consistency loss $L_{cyc}$. Figure 3: CAST consists of an encoder-transformation-decoder-based generator $G$, a multi-layer style projector (MSP) module, and a domain enhancement module. We first generate images $I_{cs}$ and $I_{sc}$ from the content image $I_c$ and the style image $I_s$ using the generator. Then, $I_{cs}$ and $I_{sc}$ are fed into the MSP module to generate the corresponding style code $z$ and $\bar{z}$, which will be used as positive samples in the style contrastive learning process. The style codes $z^-$ of other artistic images in the style bank will be used as negative samples. We compute a contrastive style loss $L_{contra}$ based on these style codes. DE module is based on the adversarial loss $L_{adv}$ and the cycle consistency loss $L_{cyc}$. Artist of style image: Giovanni Battista Piranesi.

Ulyanov et al. [2016; Zhang and Dana 2018] represent multiple styles using one single model. Arbitrary style transfer methods [Deng et al. 2022, 2020; Li et al. 2017; Svoboda et al. 2020; Wu et al. 2021a] build more flexible feed-forward architectures to handle an arbitrary style using a unified model. AdaIN [Huang and Belongie 2017] and DIN [Jing et al. 2019] directly align the overall statistics of content features with the statistics of style features and adopt conditional instance normalization. However, dynamic generation of affine parameters in the instance normalization layer may cause distortion artifacts.

Instead, several methods follow the encoder-decoder manner, where feature transformation and/or fusion is introduced into an auto-encoder-based framework. For example, Li et al. [2019] learn a cross-domain feature linear transformation matrix (LST) to enable universal style transfer and generate the desired stylization results by decoding from the transformed features. Park et al. [2019] introduce SANet to flexibly match the semantically nearest style features onto the content features. Deng et al. [2021] propose MCCCNet to fuse exemplar style features and input content features by multi-channel correlation for efficient style transfer. An et al. [2021] propose reversible neural flows and an unbiased feature transfer module (ArtFlow) to prevent content leak during universal style transfer. Liu et al. [2021b] present an adaptive attention normalization module (AdaAttN) to consider both shallow and deep features for attention score calculation. GAN-based methods [Kotovenko et al. 2019a;b; Sanakoyeu et al. 2018a; Svoboda et al. 2020; Zhu et al. 2017] have been successfully used in collection style transfer, which considers style images in a collection as a domain [Chen et al. 2021b; Lin et al. 2021; Xu et al. 2021].

Contrastive learning. Contrastive learning has been used in many applications, such as image dehazing [Wu et al. 2021b], context prediction [Santa Cruz et al. 2019], geometric prediction [Liu et al. 2019] and image translation. Contrastive learning is introduced in image translation to preserve the content of the input [Han et al. 2021] and reduce mode collapse [Jeong and Shin 2021; Kang and Park 2020; Liu et al. 2021a]. CUT [Park et al. 2020] proposes patch-wise contrastive learning by cropping input and output images into patches and maximizing the mutual information between patches. Following CUT, TUNIT [Baek et al. 2021] adopts contrastive learning on images with similar semantic structures. However, the semantic similarity assumption does not hold for arbitrary style transfer tasks, which leads the learned style representations to a significant performance drop. IEST [Chen et al. 2021a] applies contrastive learning to image style transfer based on feature statistics (mean and standard deviation) as style priors. The contrastive loss is calculated only within the generated results. Contrastive learning in IEST is an auxiliary method to associate stylized images sharing the same style, and the ability comes from the feature statistics from pre-trained VGG. Differently, we introduce contrastive learning for style representation by proposing a novel framework that uses visual features comprehensively to represent style for the task of arbitrary image style transfer.

3 METHOD

As shown in Figure 3, our framework consists of three key components: (1) a multi-layer style projector which is trained to project features of artistic image into style code; (2) a contrastive style learning module which is applied to guide both the training of the multi-layer style projector and the style image generation; and (3) a domain enhancement scheme to further help learn the distribution of artistic image domain. All these components are used for learning style representations to measure the difference between the input artistic images and generated results and thus, they could be applied to different kinds of arbitrary style transfer networks.

3.1 Multi-layer Style Projector

Our goal is to develop an arbitrary style transfer framework that can capture and transfer the local stroke characteristics and overall
appearance of an artistic image to a natural image. A key component is to find a suitable style representation which can be used to distinguish different styles and further guide the generation of style images. To this end, we design an MSP module, which includes a style feature extractor and a multi-layer projector. Instead of using features from a specific layer or a fusion of multiple layers, our MSP projects features of different layers into separate latent style spaces to encode local and global style cues.

Specifically, we adapt VGG-19 [Simonyan and Zisserman 2014] and finetune the VGG-19 model pre-trained on ImageNet with a collection of 18,000 artistic images in 30 categories. We then select M layers of feature maps in VGG-19 as input to our multi-layer projector (we use layers of ReLU1_2, ReLU2_2, ReLU3_3, and ReLU4_3 in all experiments). We use max pooling and average pooling to capture the mean and peak value of features. The multi-layer projector consists of pooling, convolution, and several multilayer perceptron layers, and it projects the style features into a set of K-dimensional latent style code, as shown in Figure 4.

After training, MSP can encode an artistic image into a set of latent style code \{z_i, i \in [1, M], z_i \in \mathbb{R}^K\}, which can be plugged into an existing style transfer network (i.e., replacing the mean and variance in AdaIN [Huang and Belongie 2017]) as the guidance for \( \mathcal{L}_\text{MSP} \). It is worth noting that we calculate the contrastive measurement for network training. Therefore, we adopt contrastive learning and design a new contrastive style loss as an implicit measurement for network training.

When train the MSP module, an image \( I \) and its augmented version \( I' \) (random resizing, cropping, and rotations) are fed into a \( M \)-layer style feature extractor, which is the pre-trained VGG-19 network. The extracted style features are then sent to the multi-layer projector, which is an \( M \)-layer neural network and maps the style features to a set of \( K \)-dimensional vectors \( \{z\} \). The contrastive representation learns the visual styles of images by maximizing the mutual information between \( I \) and \( I' \) in contrast to other artistic images within the dataset considered as negative samples \( \{I^\neg\} \). Specifically, the images \( I, I', \) and \( N \) negative samples are respectively mapped into \( M \) groups of \( K \)-dimensional vectors \( z, z^+, z^\neg \in \mathbb{R}^K \) and \( \{z^\neg \in \mathbb{R}^K\} \). The vectors are normalized to prevent collapsing. Following [Van den Oord et al. 2018], we define the contrastive loss function to train our MSP module as:

\[
\mathcal{L}_\text{contra}^{\text{MSP}} = -\sum_{i=1}^{M} \log \frac{\exp(z_i \cdot z_i^+ / \tau)}{\exp(z_i \cdot z_i^+ / \tau) + \sum_{j=1}^{N} \exp(z_j \cdot z_i^+ / \tau)},
\]

where \( \cdot \) denotes the dot product of two vectors, and \( \tau \) is a temperature scaling factor and is set to 0.07 in all of our experiments. Meanwhile, we maintain a large dictionary of 4096 negative examples using a memory bank architecture following MOCO [He et al. 2020]. It is worth noting that we calculate the contrastive loss between images, as opposed to CUT [Park et al. 2020] which adopts contrastive learning by cropping images into patches and maximizing the mutual information between patches.

The contrastive representation also provides proper guidance for the generator \( G \) to transfer styles between images. We adopt the same form of contrastive loss as used for learning MSP in Eq. (1), but compute the loss using the contrastive representations of the output image \( I_{cs} \) and the reference style image \( I_s \), then \( I_{cs} \) will have a style similar to \( I_s \):

\[
\mathcal{L}_\text{contra} = \sum_{i=1}^{M} \log \frac{\exp(\hat{z}_i \cdot \hat{z}_i^+ / \tau)}{\exp(\hat{z}_i \cdot \hat{z}_i^+ / \tau) + \sum_{j=1}^{N} \exp(\hat{z}_j \cdot \hat{z}_i^+ / \tau)},
\]

where \( \hat{z} \) and \( \hat{z}^\neg \) denote the contrastive representation of \( I_{cs} \) and \( I_s \), respectively. The negative examples are sampled from the same dictionary used for training of the MSP module. Notably, we take the specific generated and reference images as positive examples and utilize contrastive loss as guidance to transfer styles, which is an one-on-one process. Differently, the contrastive loss in IEST [Chen et al. 2021a] is calculated only within generated results and it takes a set of images as positive examples, which may reduce the style consistency with the given reference (see Figure 2).

3.3 Domain Enhancement

We introduce DE with adversarial loss to enable the network to learn the style distribution Recent style transfer models employ GAN [Goodfellow et al. 2014] to align the distribution of generated images with specific artistic images [Chen et al. 2021b; Lin et al. 2021]. The adversarial loss can enhance the holistic style of the stylization results, while it strongly relies on the distribution of datasets. Even with the specific artistic style loss, the generation process is often not robust enough to be artifact-free.
Different from these previous methods, we divide the images in the training set into realistic domain and artistic domain, and we use two discriminators $D_R$ and $D_A$ to enhance them respectively (see Figure 3). During the training process, we first randomly select an image from the realistic domain as the content image $I_c$ and another image from the artistic domain as the style image $I_s$, and $I_s$ are used as the real samples of $D_R$ and $D_A$, respectively. The generated image $I_{cs} = G(I_c, I_s)$ is used as the fake sample of $D_A$. We exchange the content and style images to generate an image $I_{sc} = G(I_s, I_c)$ as the fake sample of $D_R$. The adversarial loss is:

$$L_{adv} = \mathbb{E} [\log D_R(I_c)] + \mathbb{E} [\log (1 - D_R(I_{cs}))]
+ \mathbb{E} [\log D_A(I_s)] + \mathbb{E} [\log (1 - D_A(I_{sc}))].$$

To maintain the content information of the content image in the process of style transfer between the two domains, we also add a cycle consistency loss:

$$L_{cyc} = \mathbb{E} [||I_c - G(I_{cs}, I_s)||_1] + \mathbb{E} [||I_s - G(I_{sc}, I_c)||_1].$$

### 3.4 Network Training

Our full objective function for training of the generator $G$ and discriminators $D_R$ and $D_A$ is formulated as:

$$L(G, D_R, D_A) = \lambda_1 L_{adv} + \lambda_2 L_{cyc} + \lambda_3 L_{contra}^{G}$$

where $\lambda_1$, $\lambda_2$, and $\lambda_3$ are weights to balance different loss terms. We set $\lambda_1 = 1$, $\lambda_2 = 2$, and $\lambda_3 = 0.2$ in all of our experiments.

**Implementation details.** We collect 100,000 artistic images in different styles from WikiArt [Phillips and Mackintosh 2011] and randomly sample 20,000 images as our artistic dataset. We average sample 20,000 images from Places365 [Zhou et al. 2018] as realistic image dataset. We train and evaluate our framework on those artistic and realistic images. In the training phase, all images are loaded with $256 \times 256$ resolution. The number of feature map layers $M$ is set to be 4. The dimension $K$ of style latent code is set to 512, 1024, 2048, and 2048 for the four different layers, respectively. We use Adam [Kingma and Ba 2014] as optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$, and a batch size of 4. The initial learning rate is set to $1 \times 10^{-4}$ and linear decayed linear for total $8 \times 10^5$ iterations. The training process takes about 18 hours on one NVIDIA GeForce RTX3090. We choose the same backbone as AdaIN [Huang and Belongie 2017] in our experiments for simplicity. The results of using other backbones are shown in the supplementary materials.

### 4 EXPERIMENTS

We compare CAST with several state-of-the-art style transfer methods, including NST [Gatys et al. 2016], AdaIN [Huang and Belongie 2017], LST [Li et al. 2019], SANet [Park and Lee 2019], ArtFlow [An et al. 2021], MCCNet [Deng et al. 2021], AdaAttN [Liu et al. 2021b], and IEST [Chen et al. 2021a]. All the baselines are trained using publicly available implementations with default configurations. The comparison of inference speed is shown in Table 1.

#### 4.1 Qualitative Evaluation

We first present qualitative results of our method against the selected state-of-the-art methods in Figure 5. The comparison shows the superiority of CAST in terms of visual quality. NST is likely to encounter the issue of unpleasant local minimum (e.g., the 1st, 5th and 8th rows). AdaIN often fails to generate sharp details and introduces undesired patterns that do not exist in style images (e.g., the 1st, 3rd, 6th and 8th rows). LST tends to transfer low-level style patterns like colors but the local details of strokes are often ignored (e.g., the 2nd, 5th, 6th and 8th rows). ArtFlow sometimes generates unexpected colors or patterns in relatively smooth regions in some cases (e.g., the 1st, 5th and 6th rows). MCCNet can effectively preserve the input content but may fail to capture the stroke details and often generates haloing artifacts around object contours (e.g., the 2nd, 4th, 6th, 8th rows). AdaAttN cannot well capture some stroke patterns (e.g., the 1st, 3rd, 4th and 6th rows) and fails to transfer important colors of the style references to the results (the 2nd and 8th rows). Although the generated visual effects of IEST are of high quality, the usage of second-order statistics as a global style descriptor causes color distortion (e.g., the 1st row in Figure 2e and the 4th row in Figure 5) and cannot capture the detailed stylized patterns (e.g., the regions of sky in the 5th and 7th rows in Figure 5). In particular, these state-of-the-art methods cannot capture the leaving blank characteristic of Chinese painting style in the 1st row of Figure 5 and fail to generate results with a clean background.

In comparison, CAST achieves the best stylization performance that balances characteristics of style patterns and content structures. Instead of using second-order statistics as a global style descriptor, we use an MSP module for style encoding with the help of a DE module for effective learning of style distribution. Thus, CAST can flexibly represent vivid local stroke characteristics and the overall appearance while still preserving the content structure. For instance, as shown in Figures 1h and 2c (the 2nd row), CAST successfully captures the large portion of empty regions in the style images, and it generates a stylization results which have salient objects in the clean background.

#### 4.2 Quantitative Evaluation

We use the content loss [Li et al. 2017], LPIPS [Chen et al. 2021a], and deception rate [Sanakoyeu et al. 2018b] and conduct two user studies to evaluate our method quantitatively. The two user studies are online surveys that cover art/computer science students/professors and civil servants.

For content loss and LPIPS, we use a pre-trained VGG-19 and compute the average perceptual distances between the content image and the stylized image. The statistics are shown in Table 1. For deception rate, we train a VGG-19 network to classify 10 styles on WikiArt. Then, the deception rate is calculated as the percentage of stylized images that are predicted by the pre-trained network as the correct target styles. We report the deception rate for the proposed CAST and the baseline models in the 2nd column of Table 1. As observed, CAST achieves the highest accuracy and surpasses other methods by a large margin. As a reference, the mean accuracy of the network on real images of the artists from WikiArt is 78%.

**User Study I.** We compare CAST with eight state-of-the-art style transfer methods to evaluate which method generates results that
Content Style Ours IEST AdaAttN MCCNet ArtFlow SANet LST AdaIN NST

Figure 5: Qualitative comparisons with several state-of-the-art style transfer methods, including IEST [Chen et al. 2021a], AdaAttN [Liu et al. 2021b], MCCNet [Deng et al. 2021], ArtFlow [An et al. 2021], SANet [Park and Lee 2019], LST [Li et al. 2019], AdaIN [Huang and Belongie 2017], and NST [Gatys et al. 2016]. Artists of style images (from top to bottom): Baishi Qi, August Macke, Vincent van Gogh, Vincent van Gogh, Paul Cezanne, Vincent van Gogh, Claude Monet, and Nicholas Roerich.

are most favored by humans. For each participant, 50 content-style pairs are randomly selected and the stylized results of CAST and one of the other methods are displayed in a random order. Then, we ask the participant to choose the image that learns the most characteristics from the style image. Participants were told that the consistency of content and style was the primary metrics. The style is subjective and the effectiveness of training also depends on their understanding ability. Finally, we collect 3,400 votes from 68 participants. We report the percentage of votes for each method in the 3rd column of Table 1. CAST obtains significantly higher preferences in categories of Sketch, Chinese painting, and Impressionism.

User Study II. We design a novel user study to evaluate the stylized images quantitatively, which is called the Stylized Authenticity Detection (SAD). For each question, we show participants ten artworks of similar styles, including two to four stylized fake painting and ask them to select the synthetic ones. Within each single question, the stylized paintings are generated by the same method. Each participant finished 25 questions. Finally, we collect 2125 groups of results from 85 participants and use the average precision and recall as the measurement for how likely the results will be recognized as synthetics. Table 1 shows the statistics. The paintings generated by CAST have the lowest chance to be decided by people as fake paintings. We also notice that the precision and recall of CAST is less than 50%, which means that users could not tell the real ones from the fakes and tend to select more real paintings as synthetics when doing the testing.

4.3 Ablation Study

Contrastive style loss. We replace the contrastive style loss with Gram matrix-based perceptual loss, i.e., the model includes perceptual loss, adversarial loss, and cycle consistency loss. As shown in Figures. 6e and 6i, the model using Gram matrix instead of our contrastive style loss cannot capture the stroke characteristics of the style image compared with the full CAST model. The sharp pencil
The results of user study II show the accuracy and recall of being selected as fake paintings by the participants. The best results are in bold while the second best results are marked with underline.

**Table 1: Statistics of inference speed and quantitative comparison with state-of-the-art methods.** The results of user study I represent the average percentage of cases in which the result of the corresponding method is preferred compared with ours. The results of user study II show the accuracy and recall of being selected as fake paintings by the participants. The best results are in bold while the second best results are marked with underline.

| Method          | CAST  | IEContraAST | AdaAttN | MCCNet | Artflow | SANet | LST | AdaIN | NST  |
|-----------------|-------|-------------|---------|--------|---------|-------|-----|-------|------|
| Inference time  |       |             |         |        |         |       |     |       |      |
| (ms/image)      | 11    | 184         | 130     | 29     | 168     | 14    | 7   | 11    | 16863|
| Content loss    | 0.148 | 0.155       | 0.162   | 0.117  | 0.172   | 0.150 | 0.155| 0.176 | 0.188|
| LPIPS           | 0.245 | 0.256       | 0.256   | 0.234  | 0.264   | 0.265 | 0.248| 0.266 | 0.291|
| Deception Rate  | **62.00%** | **56.42%** | **50.70%** | **46.37%** | **43.79%** | **51.87%** | **48.29%** | **51.00%** | **37.70%** |
| User Study I    |       |             |         |        |         |       |     |       |      |
| Precision       | 43.69% | 60.26%     | **58.64%** | 70.89% | 56.81%  | 65.80% | 65.49%| 70.66% | 60.18%|
| Recall          | **41.19%** | 58.67%     | **58.16%** | 72.76% | 61.55%  | 62.55% | 64.91%| 75.93% | 64.76%|

Figure 6: Ablation study results. From left to right: (a) content image; (b) style image; (c) AdaIN with perceptual loss; (d) CAST without DE; (e) AdaIN with style loss based on Gram matrix, DE, and cycle consistency loss; (f) CAST using mixed DE; (g) CAST using one DE without the realistic domain; (h) CAST trained with asymmetric cycle consistent loss by only reconstructing the realistic images; and (i) full CAST model. Artists of style images: Master of the Small Landscapes and Alexandre Istrati.

Domain enhancement. Our full CAST uses DE for realistic and artistic images separately. We train a simplified CAST model using one discriminator that mixes realistic and artistic images together (mix-DE). As shown in Figure 6f, the results generated by mix-DE model are acceptable, but the stroke details in the generated images are weaker than the ones by the full CAST model. This fact is due to the existence of a significant gap between the artistic and realistic image domains. We further abandon all images from realistic domain for ablation. As shown in Figure 6g, the results generated by one-DE model lack details.

Cycle consistency loss. To better evaluate the improvement of the contrastive style loss on the style transfer task, we exclude the latent promotion of cycle consistency loss from network training. The reason is that the reconstruction process of artistic image may imply style information. We train CAST with an asymmetric cycle consistent loss, which only reconstructs the realistic images. The decoder of the style transfer network is unaffected by the reconstruction of the artistic image. As shown in Figure 6h, removing realistic image reconstruction will lead to slightly degraded stylization results.

5 CONCLUSION AND FUTURE WORK

In this work, we present a novel framework, namely CAST, for the task of arbitrary image style transfer. Instead of relying on second-order metrics such as Gram matrix or mean/variability of deep features, we use image features directly by introducing an MSP module for style encoding. We develop a contrastive loss function to leverage the available multi-style information in the existing collection of artwork and help train the MSP module and our generative style transfer network. We further propose a DE scheme to effectively model the distribution of realistic and artistic image domains. Extensive experimental results demonstrate that our proposed CAST method achieves superior arbitrary style transfer results compared with state-of-the-art approaches. In the future, we plan to improve the contrastive style learning process by considering artist and category information.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China (no. 2020AAA0106200), National Science Foundation of China (nos. 61832016, U20B2070, 621070958), Ministry of Science and Technology, Taiwan (no. 110-2221-E-006-135-MY3), and Open Projects Program of National Laboratory of Pattern Recognition.
