Unsupervised Bidirectional Style Transfer Network using Local Feature Transform Module

Kangmin Bae Hyung-Il Kim Yongjin Kwon Jinyoung Moon
ETRI
Republic of Korea
{kmbae, hikim, scocso, jymoon}@etri.re.kr

Abstract

In this paper, we propose a bidirectional style transfer method by exchanging the style of inputs while preserving the structural information. The proposed bidirectional style transfer network consists of three modules: 1) content and style extraction module that extracts the structure and style-related features, 2) local feature transform module that aligns locally extracted feature to its original coordinate, and 3) reconstruction module that generates a newly stylized image. Given two input images, we extract content and style information from both images in a global and local manner, respectively. Note that the content extraction module removes style-related information by compressing the dimension of the feature tensor to a single channel. The style extraction module removes content information by gradually reducing the spatial size of a feature tensor. The local feature transform module exchanges the style information and spatially transforms the local features to its original location. By substituting the style information with one another in both ways (i.e., *global and local) bidirectionally, the reconstruction module generates a newly stylized image without diminishing the core structure. Furthermore, we enable the proposed network to control the degree of style to be applied when exchanging the style of inputs bidirectionally. Through the experiments, we compare the bidirectionally style transferred results with existing methods quantitatively and qualitatively. We show generation results by controlling the degree of applied style and adopting various textures to an identical structure.*

1. Introduction

Visual imagination is one of the most remarkable aspects of human intelligence. For instance, by observing the brown loafer, humans can easily think up a certain type of bag with a brown texture. This imagination originates from humans' ability to separate the style and structure of an object independently. Inspired by the ability, there has been much progress regarding image generation algorithms due to the advance of deep generative models. One of the representative works is artistic style transfer [8, 14], which refers to merging the content of a photo and the style of a painting. Although the style transfer algorithms generated astonishing results, they mainly considered the style of artistic paintings with a distinct style. Another approach is image-to-image translation [16] that transforms an image from one domain to have the style (or characteristics) of another. After the introduction of adversarial [10] and cycle consistency loss [41], the quality of translated images has improved dramatically. However, handling the controllability (*i.e.*, the degree of the reference style to be applied) of a generation network remained unresolved.

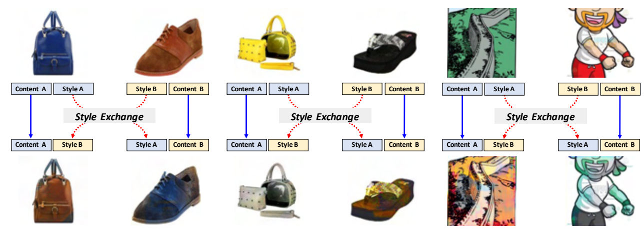

Figure 1. The generation results of the proposed bidirectional style transfer network. Based on an input pair, style information is exchanged with each other and generates newly stylized images bidirectionally.

To overcome the aforementioned limitations, we propose a bidirectional style transfer network with local feature transform module that considers the multiple inputs provided by a user, thus generating multiple outputs by exchanging styles with each other. Assuming a pair of input images as depicted in Fig. 1, our network can generate a pair of outputs that preserve the original structure and apply the style of another bidirectionally. Specifically, the proposed network consists of extraction and reconstruction modules. The extraction module predicts two features related to the style and content information from the two input images respectively. Based on the disentangled features, the reconstruction module exchanges the styles and generates new images bidirectionally (*i.e*., style-exchanged images). Furthermore, we enable the proposed network to control the effects of the style feature on the generated images by introducing a weight parameter. With Edges2Handbag [40], Edges2Shoes [16], and Clipart [33] datasets, we compared the generation results with other methods in quantitative and qualitative ways. Through the experiments, we validated that our method could exchange the style of inputs with another. Furthermore, we show that our method can apply various styles and control the degree of stylization when generating outputs.

2. Related Works

2.1. Artistic Style Transfer

Neural style transfer is referred to as an image generation method that applies the target style (*e.g*., style of artistic paintings) to the input while preserving the main structure [23]. One of the early attempts to adopt a neural network as a style transfer method was the work from Gatys *et al*. [8]. They utilized the Frobenius norm of a gram matrix as a style loss which is regarded as considering the correlation of the features. Another approach was to generate synthetic style transferred images by combining convolutional networks and a Markov Random Field (MRF) to maintain the local pattern of the style exemplar [25].

After the introduction of artistic neural style transfer from Gatys *et al*. [8], Li *et al*. [26] provided a mathematical explanation of style loss using Maximum Mean Discrepancy. To accelerate the style transfer, Johnson *et al*. [18] trained a style transfer network to synthesize images in a real-time manner. However, the work of Johnson *et al*. [18] required an additional training process when the target style is modified. Eventually, Huang *et al*. [14] proposed real-time style transfer using an adaptive instance normalization layer. While previous methods mostly focused on applying the style of artistic masterpieces, Luan *et al*. [29] proposed a method to transfer the style of a photo. For further improvements, Penhout *et al*. [34] detached a salient object from the background. This method performed style transfer of salient object and background separately to prevent any disruption occurring due to style difference of background and object. Kim *et al*. [20] considered geometric information between style and content images while applying the texture information. Liao *et al*. [27] utilized semantic context matching and applied texture information in a global way after considering the local context.

2.2. Image-to-image Translation

The image-to-image translation refers to generating an image from one domain to another. Unlike traditional computer vision problems [1, 7, 12], Isola *et al*. [16] defined image-to-image translation as a generalized representation of many previous vision tasks. In this manner, Isola *et al*. [16] proposed a generalized translation model referred to as Pix2Pix which uses adversarial [10] loss while training. For further translation, Zhu *et al*. [42] proposed BicycleGAN to perform one-to-many generation. Although the Pix2Pix [16] and BicycleGAN [42] showed impressive results, they require a pair of input and ground truth to be targeted for translation. However, after the proposal of cycle-consistency [21, 41] loss, the pair of input with ground truth was no longer required.

Hoffman *et al*. [13] proposed a simple method to obtain real-world image datasets by translating virtual images to real-world images. To obtain more user-guided images, a style removal network was used to detach all the texture information before applying a new type of texture [2, 3]. Instead of removing the original style, Ge *et al*. [9] utilized a segmentation map to generate separated regions. For further efforts, attaching a refinement network referred to as Pix2PixHD [36] was proposed to generate a higher resolution image. Since Pix2PixHD [36] considers the segmentation map as an input, it tends to wash away the information of semantic masks. Therefore, Park *et al*. [31] proposed a segmentation map-based denormalization layer that can handle the feature map without washing out the semantic information.

Zhu *et al*. [43] considered region adaptive normalization in a class-wise manner instead of regarding the whole semantic masks. For further improvements, Kim *et al*. [19] added an attention layer to the extracted feature while generating the new translated image. Another approach was to assume an intermediate domain that keeps both characteristics of domains [15, 28]. While previous methods considered two domain translation, Choi *et al*. [4, 5] proposed multiple domain image-to-image translator with the unified generator. Park *et al*. [32] proposed a swapping autoencoder with co-occurrent patch statistics to encourage texture codes to represent the texture information of generated images. Based on the above previous works, we consider an image-to-image translation network that accepts various target styles as many style transfer methods behave.

3. Methods

3.1. Bidirectional Style Transfer Network

Given two input images, we generate newly transferred outputs by exchanging the style bidirectionally. To achieve this goal, we propose a bidirectional style transfer network that generates style-exchanged images for two arbitrary input images sampled from the dataset $\mathcal{X}$. The proposed bidirectional style transfer network consists of three modules: 1) content and style feature extraction module, 2) recon-

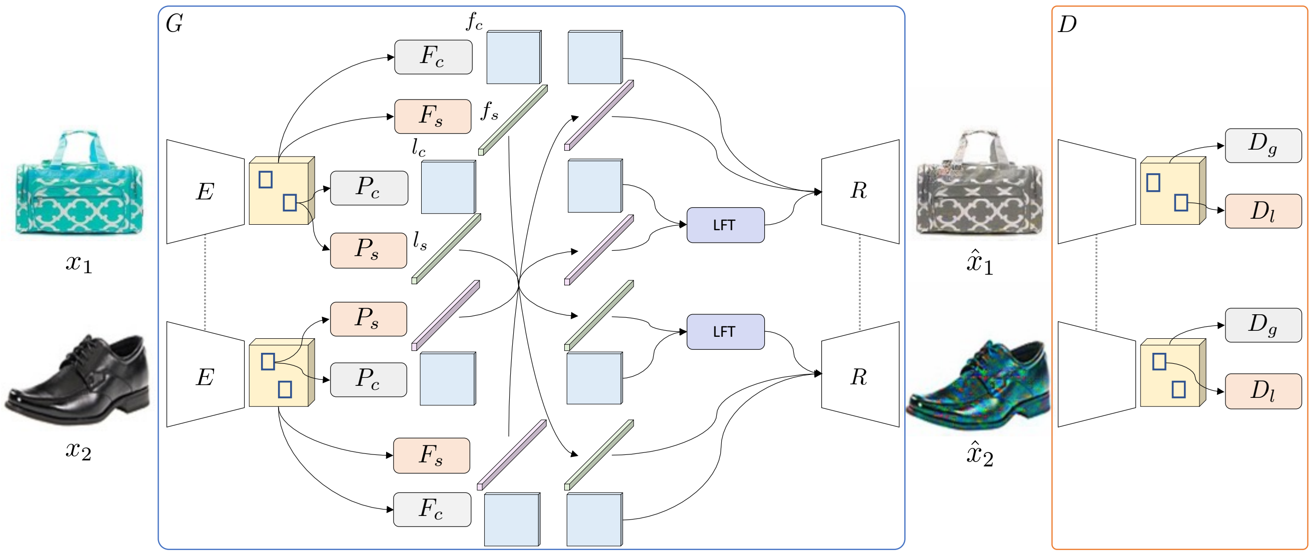

Figure 2. Network architecture of the proposed bidirectional style transfer network. The proposed network G consists of feature extractor, feature transformer and reconstructor. The dashed line denotes the weight parameters share the identical value. The content feature extraction module F_c, P_c removes style-related information and style feature extraction module F_s, P_s removes content information. The reconstruction module R generates new images by merging the content and style feature. Lastly, the discriminator D predicts the probability of whether the image is sampled from real dataset in global and local manner.

struction module and 3) Local Feature Transform (LFT) module. As depicted in Fig. 2, the feature extraction module has two separated branches for global and local feature extractions based on the shared encoder network E. The content and style feature extraction module provides features extracted from the given input images, in a global and local manner. The LFT module reallocates the local features into their original location by using a spatial transformation network. Finally, the reconstruction module generates style-exchanged images with structural consistency based on the extracted features.

For a given input pair $(x_1, x_2) \sim \mathcal{X} \times \mathcal{X}$, the shared encoder network provides feature g_1 and g_2 which are given as:

$$
\begin{aligned}
g_1 &= E(x_1), &(1)\\
g_2 &= E(x_2). &(2)
\end{aligned}
$$

Based on these features, the content and style feature extraction modules, E_c and E_s, extract global content feature f_c and style feature f_s which are given as:

$$
\begin{aligned}
f_c &= F_c(g_1), &(3)\\
f_s &= F_s(g_2). &(4)
\end{aligned}
$$

To obtain local content l_c and local style l_s feature, we adopt RoIAlign [11] layer before inferencing local-level features. The local-level features are extracted using local content and style feature extraction module P_c and P_s. The results from feature extraction modules P_c, P_s are given as:

$$
\begin{aligned}
l_c &= P_c(\text{RoIAlign}(g_1, b_1)), &(5)\\
l_s &= P_s(\text{RoIAlign}(g_2, b_2)), &(6)
\end{aligned}
$$

where $b_1, b_2 \in \mathbb{R}^4$ denotes coordinates of bounding boxes which are sampled randomly. To consider various parts of local features, we selected n RoIs while exchanging the style of reconstructed images.

The content feature extraction module consists of convolution layer preventing any downsizing computations of the output feature. Therefore, outputs from the content feature extraction module diminish all style-related details and leave only the structural information. We design f_c and l_c to be projected to a single channel space by reducing the output channel of convolution filter gradually. The style feature extraction module is composed of multiple convolution and down-sampling layers. As a result, the style features f_s and l_s preserve texture details and remove structural information. That is, style features are designed to have a small spatial size via down-sampling while ascending the channel space to a higher dimension. Therefore, the dimensionalty of the feature space is given as:

$$
\begin{aligned}
f_c &\in \mathbb{R}^{1\times H\times W}, &(7)\\
l_c &\in \mathbb{R}^{1\times h\times w}, &(8)\\
f_s, l_s &\in \mathbb{R}^{C\times 1\times 1}, &(9)
\end{aligned}
$$

where C, H, and W denote the number of channels, height and width of global feature maps and h, w denotes height and width of local feature map.

Based on the extracted features f_c from x_1 and f_s from x_2, the reconstruction module R returns newly stylized image $\hat{x}_1$, *i.e.*, image with content of x_1 and style of x_2. The newly generated images by exchanging the styles of mutual inputs are represented as:

$$\begin{aligned} \hat{x}_1 &= G(x_1, x_2) \\ &= R(f_c, f_s, \mathrm{LFT}(l_c, l_s)), \qquad (10) \\ \hat{x}_2 &= G(x_2, x_1) \qquad (11) \end{aligned}$$

where G is a composite function of the global and local feature extractor, LFT, and reconstruction module. The G generates $\hat{x}_2$, in the same manner as Eq. (10) while content and style images are exchanged with each other. The reconstruction module has concatenation layer to merge f_c, l_c and f_s, l_s to generate $\hat{x}_1$ and $\hat{x}_2$. In addition, the degree of image style transfer can be controlled by a linear combination of global and local style information $\bar{f}_s, \bar{l}_s$ for generalization as follows:

$$\begin{aligned} \bar{f}_s &= \alpha F_s(g_1) + (1-\alpha) F_s(g_2) \qquad (12) \\ \bar{l}_s &= \alpha P_s(\mathrm{RoIAlign}(g_1, b_1)) \\ &\quad + (1-\alpha) P_s(\mathrm{RoIAlign}(g_2, b_w)) \qquad (13) \end{aligned}$$

where $\alpha \in [0, 1]$ denotes a weight parameter to control the effects of style feature on the generated images. In the case of $\alpha = 1$, the style of the generated image is fully exchanged with another image. In contrast, in the case of $\alpha = 0$, the original image is regenerated without style modification. Therefore, style interpolated images $\bar{x_1}$ is obatined as:

$$\bar{x}_1 = R(f_c, \bar{f}_s, \mathrm{LFT}(l_c, \bar{l}_s)) \qquad (14)$$

Furthermore, the proposed network has a discriminator D that distinguishes whether the input is sampled from the dataset domain $\mathcal{X}$.

3.2. Local Feature Transform Module

The Local Feature Transform (LFT) module merges local content feature l_c extracted from x_1 with replaced local style feature l_s obtained from x_2. After exchanging the style feature, we can obtain merged feature m_i like:

$$m_i = R_l(l_{c_i}, l_{s_i}), \qquad (15)$$

where R_l denotes a reconstruction network for local content and style features.

The spatial transformation network [17] as depicted in Fig. 3, aligns m_i based on the corresponding bounding box location b_i. As a result, the aligned feature $\hat{m}$ is given as:

$$\hat{m} = \sum_i \mathcal{T}_{\Theta_{b_i}}(m_i), \qquad (16)$$

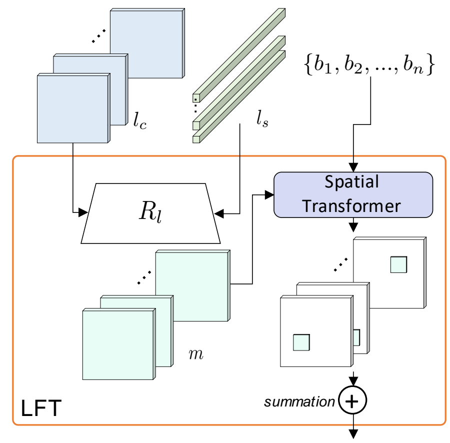

Figure 3. The architecture of Local Feature Transform (LFT) module. The LFT module generates features m_i by merging global and local features. After generating m_i, the LFT module transforms the m_i by utilizing the coordinate of bounding box b_i.

where $\mathcal{T}_{\Theta_{b_i}}$ denotes spatial transformer with parameters for scaling and translating corresponding to bounding box location b_i. After transforming the features, we added n features to represent the global information.

3.3. Training Networks

Cycle Consistency Loss The generation network G is trained by utilizing cycle consistency loss [41]. This loss enables the generation of style-transferred images based on two inputs in a cyclic manner without any ground truth supervision. The cycle consistency loss $\mathcal{L}_{cyc}$ is given as:

$$\begin{aligned} \mathcal{L}_{cyc} = \mathbb{E}_{(x_1,x_2)\sim\mathcal{X}\times\mathcal{X}}[||x_1 - G(\hat{x}_1, \hat{x}_2)||^2 \\ + ||x_2 - G(\hat{x}_2, \hat{x}_1)||^2]. \qquad (17) \end{aligned}$$

The generated image $\hat{x}_1$ preserves the original structure of the input image x_1 and obtains the main style of x_2. Likewise, the generated image $\hat{x}_2$ maintains the content of x_2 and acquires the style of the x_1. The generation network is trained by the loss that the reconstructed images $G(\hat{x}_1, \hat{x}_2)$ and $G(\hat{x}_2, \hat{x}_1)$ become the original x_1, x_2 images themselves.

Self-identity Loss When given content and style inputs are identical, the generation result should have the same structure and style as the input provided. Therefore, we devise the self-identity loss $\mathcal{L}_{sid}$ while training the generator, which is defined as:

$$\mathcal{L}_{sid} = \mathbb{E}_{x\sim\mathcal{X}}[||x - G(x, x)||^2], \qquad (18)$$

where $G(x, x)$ denotes the self-identity generation. The identity generation is the only ground truth that we can obtain during the training procedure. Through this loss, we compensate for the absence of the ground truth labels.

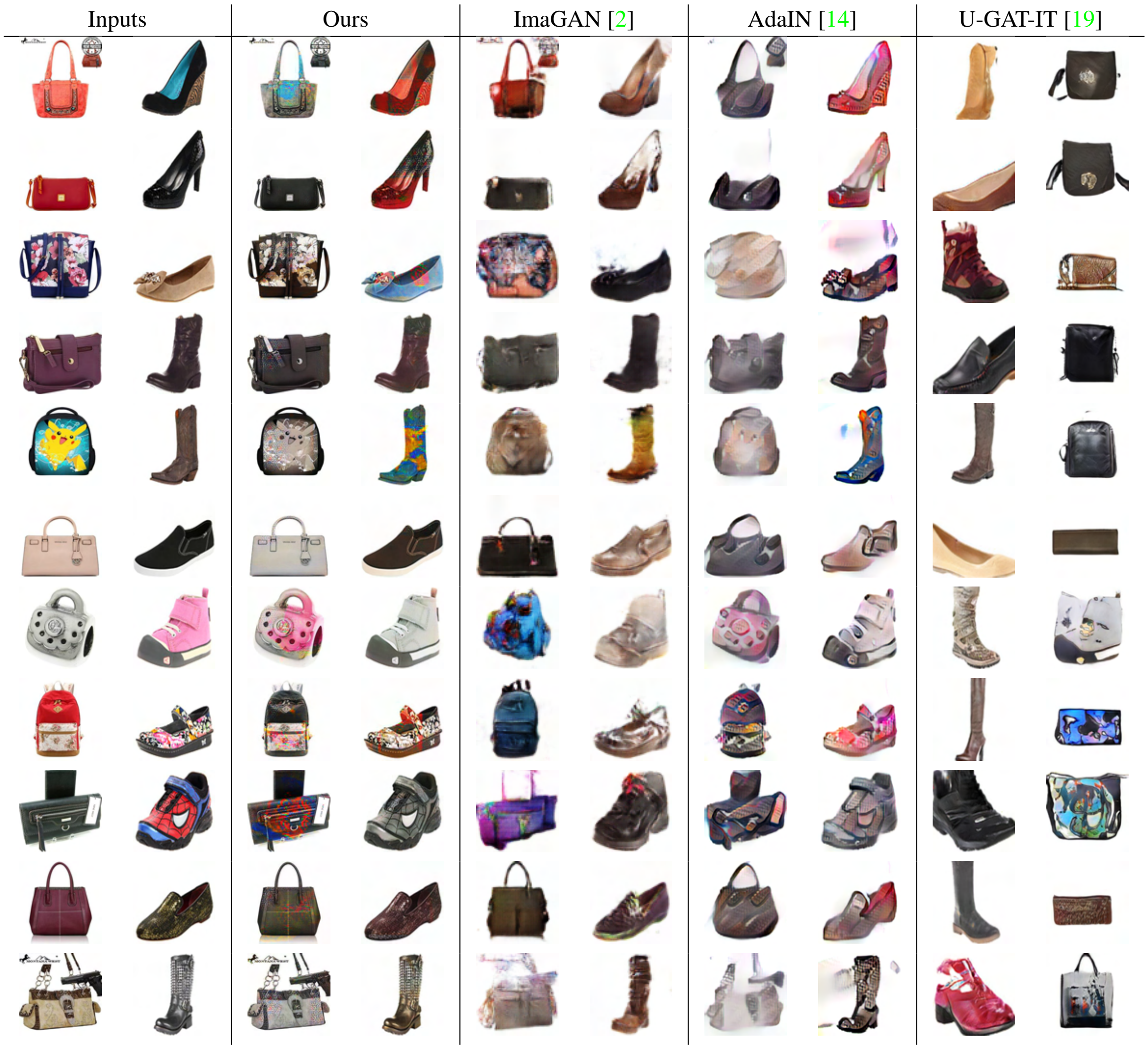

Figure 4. Qualitative results of bidirectional style transfer results of our and other methods. Our method generates two output images by exchanging the styles.

Adversarial Loss To improve the generation results, we trained the both generator and discriminator using adversarial loss [10]. With the global discriminator D_g and generator G, the global adversarial loss $\mathcal{L}_{g.adv}$ is formulated as:

$$\begin{aligned}\mathcal{L}_{g.adv} \quad = \quad & \mathbb{E}_{x\sim\mathcal{X}}[\log D_g(x)] \\ & +\mathbb{E}_{(x_1,x_2)\sim\mathcal{X}\times\mathcal{X}}[\log(1-D_g(G(x_1,x_2)))] \\ & +\mathbb{E}_{(x_1,x_2)\sim\mathcal{X}\times\mathcal{X}}[\log(1-D_g(G(\hat{x}_1,\hat{x}_2)))] \\ & +\mathbb{E}_{x\sim\mathcal{X}}[\log(1-D_g(G(x,x)))] \end{aligned} \tag{19}$$

For local discriminator, we adopt co-occurrent patch statistics [32] to induce style and content features to represent the appropriate structure and texture information. For local discriminator D_l is trained using the local adversarial loss $\mathcal{L}_{l.adv}$ which is given as:

$$\begin{aligned}\mathcal{L}_{l.adv} \quad = \quad & \mathbb{E}_{x\sim\mathcal{X}}[\log D_l(x)] \\ & +\mathbb{E}_{(x_1,x_2)\sim\mathcal{X}\times\mathcal{X}}[\log(1-D_l(G(x_1,x_2)))] \\ & +\mathbb{E}_{(x_1,x_2)\sim\mathcal{X}\times\mathcal{X}}[\log(1-D_l(G(\hat{x}_1,\hat{x}_2)))] \\ & +\mathbb{E}_{x\sim\mathcal{X}}[\log(1-D_l(G(x,x)))] \end{aligned} \tag{20}$$

Therefore the total adversarial loss $\mathcal{L}_{adv}$ is formulated as the summation of two losses which is given as:

$$\mathcal{L}_{adv} \quad = \quad \mathcal{L}_{g.adv} + \lambda_D \mathcal{L}_{l.adv}, \tag{21}$$

where λ_D denotes the weight parameter for training whole discriminator D. The adversarial loss $\mathcal{L}_{adv}$ trains discriminator D to predict the probability of the input image whether it is sampled from $\mathcal{X}$. This loss also leads the gen-

Figure 5. The test set of Edges2Shoes [16] and Edges2Handbag [40] dataset was used to generate new images. The multiple styles can be applied to a fixed structure image.

erator G to generate output $G(x_1, x_2)$ to have the same distribution as dataset $\mathcal{X}$. The generator is trained to fool the discriminator not only the generated output $G(x_1, x_2)$ but also reconstructed output $G(\hat{x_1}, \hat{x_2})$. Furthermore, the generator is also trained to fool the discriminator for the self-identity generation.

Final Objective The networks G and D are trained by using the weighted sum of loss functions introduced in the previous subsections. The total loss $\mathcal{L}_{total}$ is given as:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{cyc} + \lambda_3 \mathcal{L}_{sid}, \tag{22}$$

where λ_1, λ_2, and λ_3 are hyper-parameters that control the balance for the total loss function. The final objective is to find the optimal discriminator D^* and generator G^* via adversarial learning as follows:

$$G^*, D^* \quad = \quad \arg\min_G \max_D \mathcal{L}_{total}. \tag{23}$$

After finding G^*, we obtain a generation network that can transfer style information bidirectionally.

4. Experiments

4.1. Datasets

To train and validate our network, we used Edges2Handbag [40] and Edges2Shoes [16] datasets. The Edges2Handbag [40] dataset consists of $137k$ Amazon handbag images with 200 validation images. For Edges2Shoes [16] dataset, $50k$ images were used from UT Zappos50k [38, 39] dataset. Both fore-mentioned datasets provide edge detection results created by using HED [37] detector. In this experiment, we only utilized images without any edge detection results. Furthermore, we used the Clipart [33] dataset to validate the proposed method for complex images with many textures. The Clipart [33] dataset is comprised of $34k$ training and $14k$ test images. Among $14k$ test images, we randomly sampled 100 images and utilized them as a test dataset for further experiments.

4.2. Training Details

We trained the proposed network using Adam [22] optimizer with an initial learning rate of 0.0001, $\beta_1 = 0.5$, $\beta_2 = 0.999$ and weight decay value of 0.0001. The iteration

Figure 6. The generation results of style transferred images based on a fixed content image with warious style images. The test set of Clipart [33] dataset was used to generate new images.

persisted until it reached $20,000$ iterations, and the learning rate decay strategy was not used. We set batch size as 200 and initialized all convolution layers using LeCun initialization [24]. The input images were resized to 256×256 and normalized the pixel value. In addition, we selected $n = 8$ random region of interests while obtaining local features through RoIAlign [11]. All models were trained on Intel®Xeon®CPU E5-2640 v4 @2.40GHz with 8 Titan Xp GPUs, 256GB memory. To reach $20,000$ iterations, it took about 14 hours on our machine. For Edges2Handbag [40] and Edges2Shoes [16] datasets, the weight parameters were set to $\lambda_1 = 1$, $\lambda_2 = 10$, and $\lambda_3 = 5$. For the Clipart [33] dataset, weight parameters were set to $\lambda_1 = 2$, $\lambda_2 = 5$, and $\lambda_3 = 1$. For all experiments, we set $\lambda_D = 5$. For training stability, we adopted the least square loss instead of log-likelihood as suggested in LSGAN [30].

4.3. Qualitative Results

We show the qualitative results of our method and compared them with other algorithms. In Fig. 4, we show the generation results that the styles of two input images were transferred to each other. We observed that the proposed method generated plausible transfer results with structural consistency and style exchange. In contrast, other methods were limited in preserving the structure while exchanging the styles. Furthermore, we show the generation results according to style changes for the proposed method in Figs. 5 and 6. Even for relatively complex Clipart [33] dataset, we obtained promising results in which various styles with complex textures were transferred without disturbing the source structure. In addition, to verify the effect of the control parameter α introduced in Eq. (14), we visualized the interpolation over style feature space in Fig. 7. Note that we utilized the parameter $\alpha \in \{\frac{1}{3}, \frac{2}{3}, 1\}$ while performing the experiments for the test datasets.

4.4. Quantitative Results

We show quantitative results by comparing the top-1 and top-5 classification scores on the test set. We provide the classification accuracy score on Tab. 1. By comparing the score, we measured how the network preserved the original structural information while exchanging the style. Our method outperformed the existing methods on both top-1 and top-5 classification accuracy scores in top-1 metric. Un-

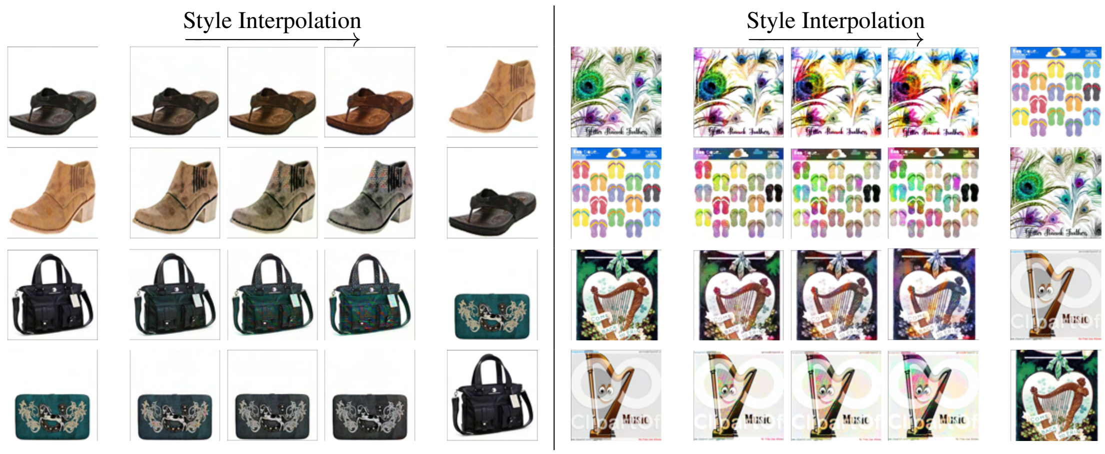

Figure 7. Qualitative results of interpolating style features with fixed value of content features. We provide bidirectional generation results of interpolation over style feature with respect to the weight parameter α for Edges2Shoes [16], Edges2Handbag [40], and Clipart [33] datasets.

	Top-1	Top-5
Original dataset [16]	62.0%	89.0%
Ours	**61.1%**	**81.2%**
ImaGAN [2]	54.0%	80.5%
DiscoGAN [21]	51.0%	77.0%
U-GAT-IT [19]	49.5%	78.5%
CycleGAN [41]	49.0%	79.5%
AdaIN [14]	37.5%	65.5%

Table 1. The comparisons of top-1 and top-5 classification performances for generated results with structure of Edges2Shoes [16] with style of Edges2Handbag [40].

like any other methods, our network nearly achieved the score of the original structures. To measure the score, we exchanged the style of test sets from Edges2Shoes [16] with the style of test samples from Edges2Handbag [40]. The classification score of generation results was calculated using Inception V3 [35] network which was already trained using ILSVRC [6] dataset. After obtaining the classification score, we calculated top-1 and top-5 scores for all algorithms. Based on the class labels of ILSVRC [6] dataset, we regarded as correct generation results when the Inception V3 [35] network predicts the class related to shoe. Compared to other methods, ours achieved competitive classification scores. Furthermore, our method showed competitive results even compared to real dataset.

5. Conclusions

We proposed a bidirectional style transfer network that accepts two inputs and generates two style transferred results. The network consists of three modules; 1) style and content feature extraction module, 2) local Feature Transform module, and 3) reconstruction module. The style and content feature extraction modules extract two features with different sizes. The content feature $f_c \in \mathbb{R}^{1\times H\times W}, l_c \in \mathbb{R}^{1\times h\times w}$ with a single channel preserves structural information while removing style-related information. The style feature $f_s, l_s \in \mathbb{R}^{C\times 1\times 1}$ with a single spatial size removes content information while preserving the textures. The LFT module transformed local features to align with its global information. The reconstruction module generates a newly stylized image by combining the content features with exchanged style information provided from the extraction module. We tested our network on Edges2Shoes [16], Edges2Handbag [40] and Clipart [33] datasets. We compared our results with other methods qualitatively and showed one-to-many generation based on a single content image. We expect our network to motivate many designers when choosing the appropriate texture of an object.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2014-3-00123, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis, No.2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network, and No.2022-0-00124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities).

References

[1] William T. Freeman Alexei A. Efros. Image Quilting for Texture Synthesis and Transfer. In *SIGGRAPH*, 2001. 2

[2] Kang Min Bae, Minuk Ma, Hyunjun Jang, Minjeong Ju, Hyoungwoo Park, and Chang D. Yoo. ImaGAN: Unsupervised Training of Conditional Joint CycleGAN for Transferring Style with Core Structures in Content Preserved. In *ACCV*, 2018. 2, 5, 8

[3] Huiwen Chang, Jingwan Lu, Fisher Yu, and Adam Finkelstein. PairedCycleGAN: Asymmetric Style Transfer for Applying and Removing Makeup. In *CVPR*, 2018. 2

[4] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *CVPR*, 2018. 2

[5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *CVPR*, 2020. 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 8

[7] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (TOG)*, 2006. 2

[8] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *CVPR*, 2016. 1, 2

[9] Chongjian Ge, Yibing Song, Yuying Ge, Han Yang, Wei Liu, and Ping Luo. Disentangled Cycle Consistency for Highly-realistic Virtual Try-On. In *CVPR*, 2021. 2

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NeurIPS*, 2014. 1, 2, 5

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 3, 7

[12] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. Image analogies. In *SIGGRAPH*, 2001. 2

[13] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. CyCADA: Cycle Consistent Adversarial Domain Adaptation. In *ICML*, 2018. 2

[14] Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *ICCV*, 2017. 1, 2, 5, 8

[15] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal Unsupervised Image-to-image Translation. In *ECCV*, 2018. 2

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image Translation with Conditional Adversarial Networks. In *CVPR*, 2017. 1, 2, 6, 7, 8

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial Transformer Networks. In *NeurIPS*, 2015. 4

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV*, 2016. 2

[19] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. U-GAT-IT: Unsupervised Generative Attentional Networks with Adaptive Layer-Instance Normalization for Image-to-Image Translation. In *ICLR*, 2020. 2, 5, 8

[20] Sunnie S. Y. Kim, Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Deformable Style Transfer. In *ECCV*, 2020. 2

[21] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In *ICML*, 2017. 2, 8

[22] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015. 6

[23] Jan Eric Kyprianidis, John Collomosse, Tinghuai Wang, and Tobias Isenberg. State of the "art": A taxonomy of artistic stylization techniques for images and video. *IEEE Transactions on Visualization and Computer Graphics*, 19(5):866–885, 2013. 2

[24] Yann Lecun, Leon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient BackProp. In *Neural networks: Tricks of the trade*, 1998. 7

[25] Chuan Li and Michael Wand. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. In *CVPR*, 2016. 2

[26] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying Neural Style Transfer. In *IJCAI*, 2017. 2

[27] Yi-Sheng Liao and Chun-Rong Huang. Semantic Context-Aware Image Style Transfer. *IEEE Transactions on Image Processing*, 31:1911–1923, 2022. 2

[28] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised Image-to-Image Translation Networks. In *NeurIPS*, 2017. 2

[29] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep Photo Style Transfer. In *CVPR*, 2017. 2

[30] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. In *ICCV*, 2017. 7

[31] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic Image Synthesis with Spatially-Adaptive Normalization. In *CVPR*, 2019. 2

[32] Taesung Park, Jun-Yan Zhu, Oliver Wang, Jingwan Lu, Eli Shechtman, Alexei A. Efros, and Richard Zhang. Swapping Autoencoder for Deep Image Manipulation. In *NeurIPS*, 2020. 2, 5

[33] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment Matching for Multi-source Domain Adaptation. In *CVPR*, 2019. 2, 6, 7, 8

[34] Sebastian Penhouët and Paul Sanzenbacher. Automated Deep Photo Style Transfer. *ArXiv*, 2019. 2

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *CVPR*, 2016. 8

[36] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *CVPR*, 2018. 2

[37] Saining Xie and Zhuowen Tu. Holistically-Nested Edge Detection. In *ICCV*, 2015. 6

[38] Aron Yu and Kristen Grauman. Fine-Grained Visual Comparisons with Local Learning. In *CVPR*, 2014. 6

[39] Aron Yu and Kristen Grauman. Semantic Jitter: Dense Supervision for Visual Comparisons via Synthetic Images. In *ICCV*, 2017. 6

[40] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative Visual Manipulation on the Natural Image Manifold. In *ECCV*, 2016. 2, 6, 7, 8

[41] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *ICCV*, 2017. 1, 2, 4, 8

[42] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward Multimodal Image-to-image Translation. In *NeurIPS*, 2017. 2

[43] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. SEAN: Image Synthesis With Semantic Region-Adaptive Normalization. In *CVPR*, 2020. 2