Artifact-Free Image Style Transfer by Using Feature Map Clamping

HYEONJIN LEE AND HYUN-CHUL CHOI
Department of Electronic Engineering, Yeungnam University, Gyeongsan 38541, South Korea
Corresponding author: Hyun-Chul Choi (pogary@ynu.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning under Grant NRF-2018R1C1B6004056.

ABSTRACT Style transfer is an application that applies colors and patterns of a style image to a content image. In the previous style transfer method, a decoder was trained by using only positive valued feature maps from the last rectified linear unit (ReLU) layer of an encoder network. Then, the trained decoder was used to generate a stylized image from a transformed feature map, which may have unseen negative values due to the transformer operation, and this resulted in odd colors and patterns on the output stylized image. In this paper, we propose a simple but effective technique, called the feature map clamping method, which eliminates negative values from a transformed feature map to resolve style degradation in output stylized image. Our experiments that compare the output style qualities of the previous and our methods verified that our method removed odd colors and patterns appearing in the previous method and that our method had 12.07% lower averaged feature loss and 7.22% higher averaged user preference than the previous method without losing computational efficiency.

INDEX TERMS Image style transfer, feature map clamping, computer vision, deep learning.

I. INTRODUCTION Style transfer that applies the colors and patterns of a style image to a content image using a neural network is an interesting topic in the computer vision research field. Since the neural style [1] based on a pixel-wise optimization technique had been proposed, various style transfer methods, such as fast feed-forward [2], [3] and arbitrary style transfer [4]–[6], have been developed by modifying network structure, loss for train, or type of transformer.

Most recent fast & arbitrary style transfer networks [4]–[6] consist of a single (or multiple) encoder network that has the structure of convolutional neural network (CNN) [13], a single (or multiple) decoder network which has the symmetrical structure with the encoder, and a single (or multiple) transformer(s) located between the encoder and decoder networks.

The first step of style transfer with the neural networks is to extract feature maps of content and style images using the encoder network. Then, the transformer changes the value of one of the following: (1) the mean and standard deviation [4], (2) covariance [5], or (3) distribution [6] of the content feature map into the style feature map. Lastly, the decoder network generates a stylized image from the transformed content feature map.

Universal style transfer [5] trained its decoder network both to reconstruct training images from their encoded feature maps and to preserve their content feature maps. The encoded feature maps always have positive values (blue bar) as shown in fig.1(a) because they came from the rectified linear units (ReLU) [12] of the encoder network. Therefore, the decoder network only recognizes positive input values from the whole training process.

However, in the test process as shown in fig.1(b), the feature map from the transformer may have negative values (red bar) and, with this unseen negative inputs, the decoder network invokes odd patterns (red blob in the bottom left region of the rightmost image of fig.1(b)) in the output stylized image.

In this paper, we propose a very simple but effective technique, called the feature map clamping method, that solves sign mismatch of feature map values between the train and test phases in the previous style transfer method. Figure 2 presents the multi-layer stylization network structure [5]
(black boxes) with our feature map clamp modules (red boxes), which removes negative values from the transformed feature map of transformer (WCT) [5]. Once our feature map clamp modules are added, only a positive valued feature map is input into the decoder in both train and test phases and no unseen negative value is input to the decoder.

II. FEATURE MAP CLAMPING

In this section, we describe the details of our feature map clamping method based on the previous model [5]. The previous model [5] has a structure of cascade networks as shown in fig.2 (except clamp modules of red boxes) where a single network consists of an encoder, a decoder, and a whitening and coloring transformer (WCT) [5]. Here, \( I_c \) and \( I_s \) represent content and style images respectively, and \( I_r \) (\( X = 1,2,3,4,5 \)) represents the output stylized image of every single network. The previous style transfer [5] using these networks utilizes the following steps to create a stylized image. Firstly, the encoder extracts two feature maps from a content image and a style image. Secondly, the transformer changes mean and covariance values from the content feature map \( f_c \) into those of the style feature map \( f_s \). Lastly, the decoder generates a stylized image from the transformed feature map. These three steps are applied consecutively to every single network to result in the final stylized image.

Since the encoders are pre-trained networks and transformer does not have any parameters to optimize in the training phase of universal style transfer [5], only the decoders are trained both to reconstruct input image and to preserve its content feature. Here, the train loss \( L_t \) (eq.1) as an objective function is calculated as a summation of reconstruction loss [11] and feature loss [2], [11].

\[
L_t = L_r + \lambda L_f,
\]

where \( \lambda \) is the relative weight of \( L_f \) to \( L_r \). Reconstruction loss \( L_r \), the first term in eq.1, is calculated as a normalized L2 distance (eq.2) between input image \( I_i \) and reconstructed output image \( I_r \) as shown in fig.3.

\[
L_r = \| |I_i - I_r| \|_2^2,
\]

Feature loss \( L_f \), the second term in eq.1, is calculated as a normalized L2 distance (eq.3) between input feature map \( \Phi(I_i) \) of each layer and the reconstructed output feature map \( \Phi(I_r) \) as shown in fig.3.

\[
L_f = \| |\Phi(I_i) - \Phi(I_r)| \|_2^2,
\]

Here, the encoded feature map \( \Phi(\cdot) \) has only positive values because the last layer of encoder network consists of rectified linear units (ReLU) and, therefore, in the training process (fig.1(a)), the decoder is trained only with positive valued inputs.

When generating a stylized image after decoder training in the previous method [5], two encoded feature maps \( (f_c, f_s) \) of input content and style images \( (I_c, I_s) \) pass through WCT transformers as shown in fig.2 which were not in the train process (fig.1(a)). The first step of WCT, whitening transform, normalizes input content feature map \( f_c \) by using its mean and covariance as eq.4.

\[
\hat{f}_c = E_c D_c^{-\frac{1}{2}} E_c^T (f_c - m_c).
\]

Then, coloring transform manipulates the normalized content feature map \( \hat{f}_c \) to have the same mean and covariance of the style feature map \( f_s \) as eq.5.

\[
\hat{f}_{cs} = E_s D_s^{-\frac{1}{2}} E_s^T \hat{f}_c + m_s,
\]

where \( m_c \) and \( m_s \) represent means, \( D_c \) and \( D_s \) eigenvalue matrices, \( E_c \) and \( E_s \) eigenvector matrices which are derived.
by eigenvalue decomposition (EVD) of feature maps $f_c$ and $f_s$ respectively.

Here, the transformed feature map from WCT may have negative values as fig.1(b) due to the whitening (eq.4) and coloring (eq.5) transform process. If these unseen negative values pass through the decoder, odd colors or patterns may appear on the output stylized image as shown as the red blob on the rightmost image of fig.1(b).

To solve this problem, we add clamp modules (red boxes in fig.2) between the transformers and decoders that reset negative values to zeros as eq.6 so that unseen negative values do not pass through the decoder.

$$\text{Clamp}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Figure 3 shows how our clamp module operates in a single style transfer network. By eliminating the negative values from the transformed feature map, we successfully remove odd patterns from the output stylized images and improve output style quality.

### III. EXPERIMENTS

As a baseline network, universal style transfer network [5] (fig.2) was used for our experiments. To mention key features, VGG-19 [8] feature extractor pre-trained with IMAGENET [7] dataset was used as the encoder networks, the symmetric structures of the encoder networks were used as the decoder networks, and WCT was used as the transformers. Importantly, for our method, clamp modules were added between transformers and decoders.

For the dataset, MS-COCO train2014 [9] including 80,000 images were used in the decoder training phase and MS-COCO test2014 [9] and Painter-By-Numbers [10] were used as content and style images respectively in the test phase. For pre-processing of training data, to prevent boundary pixel defects in the output image, we resized the shorter side of images into 512 pixels and then randomly cropped them into $256 \times 256$ pixels maintaining aspect rate. Pre-processing of test data was prepared the same way as the pre-processing of training data, except we center-crop into $256 \times 256$ pixels instead of random-crop to fairly compare the output images.
of the previous method and ours without variation in the input image region.

For training parameters, we used Adam optimizer with batch size 8 and learning rate $10^{-4}$ to minimize $L_t$ (eq.1). All experiments were operated with Pytorch v1.1.0 framework, CUDA v10.1, cudnn v7.6.2 on a single NVIDIA GTX 1080 TI device.

To compare the performances of the previous method [5] and ours, we analyzed the quality of stylized images, dark stylized images, RGB histograms of style image, feature map histogram, feature loss, and user study on each method. From here on, we refer to the previous method [5] as clamp false and the proposed method as clamp true.

As the first experiment, we qualitatively compared clamp false and clamp true by the quality of stylized images. The image set on the left side of fig.4 represent the examples of content, style, and output stylized images. From the top, two rows as a set correspond to one stylization example, where the top and bottom row represents clamp false and clamp true respectively. For each stylization example set, the far-left image on the top row represents the content image, the far-left image on the bottom row represents the style image. Subsequently, images placed to the right are the stylized images corresponding to the number of cascaded networks. We find from fig.4 that odd patterns appear in the previous method (clamp false). For example, fig.4(a) shows red colored pixels and flashing patterns on lower left corner, fig.4(c) shows red or blue colored pixels on lower right corner, fig.4(e) shows red colored pixels on cheek, and fig.4(g) shows blue colored pixels on shoulder. These odd patterns do not exist in the style images but appeared in the stylized images in clamp false. In contrast, there are no odd colors or patterns in the stylized images of our method (clamp true, fig.4(b,d,f,h)) because clamp modules (red boxes
in fig.2) removed the negative values from the transformed feature map.

Since the difference between the stylized images of clamp false and true for the dark style image (fig.4(g,h)) is not as clear as that for the bright style images (fig.4(a-f)), we adjusted the gamma value of the output stylized images for dark style images to reveal any hidden artifact. Figure 5 and fig.6 show the content, style, and the gamma-corrected stylized images for the dark style images along the cascade networks for clamp false (blue box) and clamp true (red box). We can see an odd pink pattern on the center of the images in fig.5 clamp false and an odd red pattern on the bottom center of the images in fig.6 clamp false. These artifacts are unclear when the output stylized images are dark (γ = 0.5 or 1.0) but become clearer as the gamma (γ) value increases (γ = 1.5 or 2.0). In contrast, there are only a few artifacts in clamp true (fig.5, fig.6 red box) even with high gamma values. Therefore, even though some artifacts in a stylized image are hidden because of its darkness, artifact exists in clamp false both in the cases of bright and dark style images. Those are removed by the addition of our clamp modules (red boxes in fig.2).

However, we could not find any artifact from some stylized images as shown in fig.7. We analyzed the characteristics of the style images to find the artifact appearing conditions and found that the RGB color histogram of style image is the key
factor in creating artifacts. The graphs on the right side of fig.4 and fig.7 present RGB histograms of the corresponding style images. The histograms in fig.4 that have very obvious artifacts in the stylized images show very close and high peaks. In contrast, the histograms of the style images in fig.7 with a minimum artifact in the stylized images show relatively smooth and different distributions.

As the second experiment, we compared the pixel value distributions of the transformed feature map in the test process. Figure 8 shows the first and the second channel histograms of the transformed feature map from the first network’s transformer of the cascade networks (fig.2) in case of fig.4 (a,b). While there are many negative pixel values (red bars) in clamp false (fig.8(a)), there are no negative values in clamp true (fig.8(b)), maintaining positive pixel values (blue bars). A similar trend is shown in the other channel histograms as well. The only difference between the previous method and ours is the clamp module.

Therefore, we can conclude that the addition of our clamp module is the sole reason for eliminating odd colored pixels by removing negative values from the transformed feature map.

As the third experiment, we quantitatively compared feature loss $L_f$ to evaluate the performance of each method. Here, $L_f$ is calculated as the normalized L2 distance (eq.3) between the feature map ($\Phi(I_o)$) of the output stylized image and the feature map ($\hat{f}_{cs}$) from the clamp false or true. Table 1 presents $L_f$ and the relative difference rate across 1000 result images. Table 1(a) shows the feature losses for clamp false which uses WCT transformed feature map without clamp module in calculating feature loss (fig.3 blue box) and table 1(b) for clamp true which uses the clamped feature map (fig.3 red box). Here, $L_f$ tends to increase as the number of cascade networks increases because the output image is heavily stylized. But clamp true (table 1(b)) shows comparably lower losses (table 1(c)) when compared to those of clamp
false (table 1(a)). Additionally, clamp true has an average of 12.07 % lower loss than that of the clamp false.

For the last experiment, we performed a user study with the output stylized images. We showed eight content/style image pairs and their output stylized images (total of 40 stylized images for 5 cascade networks) to 45 users, where clamp true and false show clear difference between their stylized images as fig.4. The users were asked to select a better image out of the two stylized images (one from clamp true and the other from clamp false) per cascade network. Table 2 presents preference rates from the users for each cascade network. In table 2, for the output images from the first network (second column in table 2), the preference of our method (fig.2) is 0.54 % lower than that of the previous method. Since the stylizing effect of the first network is very low (second column of fig.4), it is difficult to recognize the style quality difference from these output images. However, our method had an average of 7.22 % higher preference for output images from multiple networks (3rd, 4th, 5th, and 6th columns in table 2) where their styles are strong and clear as shown in the output images (fig.4).

For the computational cost, since the clamp module is much light-weighted compared to the convolution module in CNN, adding clamp modules increased a single network processing time (about a millisecond on NVIDIA 1080 Ti GPU without clamping module) by just several microseconds.

IV. CONCLUSION

In this paper, we proposed the feature map clamping method to remove odd colors and patterns from the output stylized image of the previous method. Our clamp module removed negative values from the transformed feature map so that no unseen negative value can be input to the decoder. Our experiments have verified that style image, of which RGB histogram has high and very close peaks, can produce artifacts on the output stylized image and that the addition of the clamp module just after feature transformer eliminates odd colors and patterns from the output stylized image. And our method achieved an average of 12.07 % lower feature loss and an average of 7.22 % higher user preference when compared to the previous method at the almost same computational cost.

REFERENCES

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2414–2423, doi: 10.1109/cvpr.2016.265.
[2] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Oct. 2016, pp. 694–711, doi: 10.1007/978-3-319-46475-6_43.

[3] V. Dumoulin, J. Shlens, and M. Kudlur, “A learned representation for artistic style,” 2016, arXiv:1610.07629. [Online]. Available: http://arxiv.org/abs/1610.07629

[4] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1501–1510, doi: 10.1109/iccv.2017.167.

[5] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, “Universal style transfer via feature transforms,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2017 pp. 386–396.

[6] L. Sheng, Z. Lin, J. Shao, and X. Wang, “Avatar-net: Multi-scale zero-shot style transfer by feature decoration,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 8242–8250, doi: 10.1109/cvpr.2018.00860.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255, doi: 10.1109/cvpr.2009.5206848.

[8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, P. Dollár, D. Ramanan, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[10] K. Nichol. (2016). Kaggle Dataset: Painter by Numbers. [Online]. Available: https://www.kaggle.com/c/painter-by-numbers

[11] A. Dosovitskiy and T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2016, pp. 658–666.

[12] V. Nair and G. E. Hinton, “Rectified linear units improve restricted Boltzmann machines,” in Proc. 27th Int. Conf. Mach. Learn. (ICML), 2010, pp. 807–814.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2012, pp. 1097–1105.

HYEONJIN LEE received the B.S. degree in electronic engineering from Yeungnam University, South Korea, in 2019, where she is currently pursuing the M.S. degree. Her research interest is computer vision techniques based on machine learning, such as image style transfer, character detection, and object segmentation.

HYUN-CHUL CHOI received the B.S. degree in electronic and electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2002, and the M.S. and Ph.D. degrees in electronic and electrical engineering from the Pohang University of Science and Technology (POSTECH), South Korea, in 2005 and 2011, respectively. From 2011 to 2014, he was a Research Scientist with the Multimedia Development Laboratory of Daum Communications. He had been an Assistant Professor with the Electronic Engineering Department, Yeungnam University, South Korea, from 2014 to 2020, where he has been an Associate Professor, since 2020. His research interests include multimedia retrieval, computational photography, 3D photogrammetry, object tracking and recognition based on machine learning, and neural networks.