Unseen Style Transfer Based on a Conditional Fast Style Transfer Network

Keiji Yanai
Department of Informatics, The University of Electro-Communications, Tokyo
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585 Japan
yanai@cs.uec.ac.jp

Abstract

In this paper, we propose a feed-forward neural style transfer network which can transfer unseen arbitrary styles. To do that, first, we extend the fast neural style transfer network proposed by Johnson et al. (2016) so that the network can learn multiple styles at the same time by adding a conditional input. We call this as "a conditional style transfer network". Next, we add a style condition network which generates a conditional signal from a style image directly, and train "a conditional style transfer network with a style condition network" in an end-to-end manner. The proposed network can generate a stylized image from a content image and a style image in one-time feed-forward computation instantly.

1 Introduction

A method on neural style transfer proposed by Gatys et al. (2015; 2016) synthesizes an image which has the style of a given style image and the content of a given content image using a CNN. However, since their method requires both forward and backward computation iteratively to synthesize a stylized image, the processing time tends to be longer (several tens of seconds) even using a GPU. Then, to accelerate it, several works using feed-forward style transfer networks which require only one-time feed-forward computation to realize style transfer have been published so far (Johnson et al. (2016); Ulyanov et al. (2016a)). Johnson et al. (2016) proposed a perceptual loss to train a ConvDeconv-style network as a feed-forward style transfer network. Their network can generate a stylized image for a given content image regarding a fixed pre-trained style in real-time.

Although Johnson et al.'s feed-forward network can treat only one fixed style, recently Dumoulin et al. (2016) proposed a method to learn multiple styles with a ConvDeconv-style fast style transfer network. They used Instance Normalization (Ulyanov et al. (2016b)) instead of Batch Normalization (Ioffe & Szegedy (2015)) for normalization of activation signals, and they proposed to replace scale and bias parameters of instance normalization layers depending on the styles. They call this as "conditional instance normalization". Although they showed that the fast style network where all the batch normalization layers were replaced with the conditional instance normalization layers had ability to learn 32 artistic styles at the same time, the transferable styles are limited to trained styles and their mixtures.

More recently, a fast arbitrary style transfer method which can transfer even untrained styles has been proposed by Chen & Schmidr (2016). They obtained feature map activations of a given content image and a given style image by VGG16, modify the feature maps by swapping each content activation patch with its closet-matching style patch, and generate a stylized image using the pre-trained inverse network which reconstructs a stylized image from the feature maps of the swapped activations. Their method is much faster than the original method by Gatys et al. (2015). However, it takes more than one second to generate a stylized image, since style swapping is a little bit complicated processing.

In this paper, we propose a feed-forward network for arbitrary style transfer. Our objective is the same as Chen & Schmidr (2016). However, our approach is different from theirs. First we extend the fast neural style transfer network proposed by Johnson et al. (2016) by adding a conditional input so that it can learn multiple styles. We call this as "a conditional style transfer network" which is similar to "a network with conditional instance normalization" (Dumoulin et al. (2016)) in terms of its objective. However, our network is simpler than theirs. While they have to prepare scale and shift parameters for all the instance normalization layers and for all the trained styles in the style transfer network, we add and connect a conditional input to the intermediate layer of the ConvDeconv-style network.

In both method, mixing of trained multiple styles is possible by providing mixed conditional weights of the different styles. From these characteristics of the conditional style transfer, we came up with the idea that training of many styles and mixing of them might bring arbitrary style transfer. To do that, a conditional network is suitable, since it can accept a real-value conditional input the dimension of which can be fixed regardless of the number of training styles.

After many trials, we found that it was possible by connecting a network which generates a conditional signal from a style image directly to the conditional input of a conditional style transfer network. We trained this conditional style transfer network with a style condition generator network in an end-to-end manner, and obtained promising results.

The basic idea for arbitrary style transfer is different from Chen & Schmidr (2016). The proposed architecture is simpler than theirs, since the network is trained in an end-to-end manner and generates a stylized image from a content image and a style image directly by one-time feed-forward computation without surgery of activation signals such as style swapping.

In this paper, we report a conditional fast neural style transfer network, and its extension for unseen style transfer.

2 Conditional fast style transfer network

We modified the ConvDeconv-style network used in Johnson et al. (2016) by adding a style condition input and an additional 1×1 convolutional layer for fusing of feature map activations and a conditional signal as shown in Figure 1. This network is inspired by Iizuka et al's CNN-based colorization work Iizuka et al. (2016). They proposed adding a scene contextual stream to a ConvDeconv-style colorization network. They transformed a scene vector, which is an output of the scene recognition network, to a feature map by making the same size of copies as the activation feature map of the main network, and concatenate it with the intermediate feature map. We followed this for adding a conditional input.

For training of the network, we prepare s style images and fifty thousands of content sample images (we used MS-COCO images in the same way as Johnson et al. (2016)), and make a mini-batch with the combinations of one content training image and all the style training images. That is, one mini-batch contains s combinations for one content image. This can be regarded as multiple style version of Instance Normalization (Ulyanov et al. (2016b)). In Dumoulin et al. (2016), they used different BN paramenters for each of the training styles, while we use common BN parameters. We guess that by putting all the styles in the same mini-batch, BN parameters are averaged over all the styles, and the conditional network is expected to be trained so that residuals from the averaged BN parameters are compensated depending on the selected style.

To train the network, we provide a content image and s-dimensional one-hot conditional vectors. As shown in Figure 1, for example, a conditional vector is set as $(1, 0, 0, ...)$ for the style no.1, while it is set as $(0, 1, 0, ...)$ for the style no.2. As a loss function, we use a perceptual loss proposed by Johnson et al. (2016). In the same way, we use VGG-16 (Simonyan et al. (2015)) as a loss network, and optimize the weights of the network so that the content feature of the output image extracted from the CONV3_3 layer of VGG-16 becomes closer to the one of an input content image and the style features (i.e. Gram matrix) of the output image extracted from the CONV1_2, CONV2_2, CONV3_3 and CONV4_3 layers become closer to the ones of the corresponding style image.

The proposed network can mix multiple styles like Dumoulin et al. (2016), although the network is trained with each of the training styles independently. The proposed network can also transfer

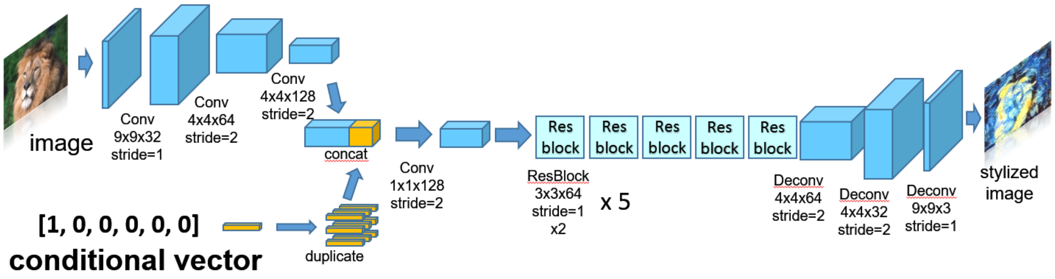

Figure 1: A conditional fast style transfer network.

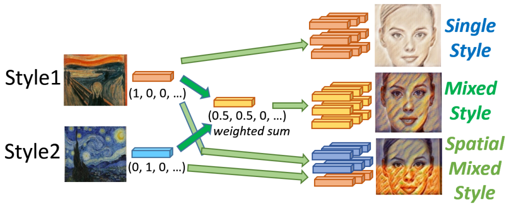

Figure 2: Three different ways to create a conditional input: (1) single style (2) mixed style by linear-weighting (3) spatially mixed style.

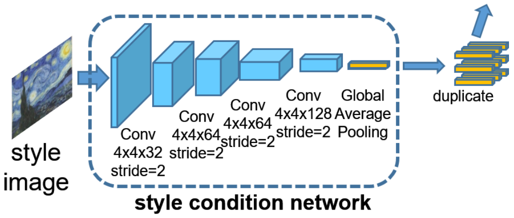

Figure 3: A style condition network which is connected to the conditional input of a conditional fast style transfer network.

different styles to the different parts of a given image at the same time, which we call "spatial style transfer" as shown in Figure 2. In the experiments, we confirmed that no quality degradation occurred in the multi-style network compared to the single network when the number of styles is not large (around 15), and linear-weighted multi-style fusion enabled us to generate various kinds of new styles which are different from the trained single styles. This observation inspired the idea of the next network.

In the experiments, we used one-hot vectors for conditions. However, a conditional signal is not limited to a binary vector. It can be a real-value vector, if it reflects "style embedding". In fact, a real-value vector which is a PCA-compressed style feature derived from a Gram matrix of VGG-16 feature maps worked. By using a real-value style condition, we can train unlimited number of style images. This property is superior to the conditional instance normalization method (Dumoulin et al. (2016)), which can treat only finite number of styles.

3 Conditional Style Transfer with a Style condition network

In the previous network, conditional signals are assumed to be given by a user. Instead, we add a CNN which takes a style image as an input and outputs a conditional signal as shown in Figure 3. By adding this style condition generator network to the previous network, the whole network can learn unlimited number of styles. To realize unseen style transfer, we expect that this CNN has ability to generate a conditional style signal for a unseen style by combining the conditional signals of the trained styles.

Note that we found that it brought better results when the last layer of the style condition network was a global average pooling (GAP). This is consistent with the fact indicated by Li et al. (2017), which is that per-channel mean and variance also represent the statistics of image styles.

We train this network in the similar way to training of the normal conditional style network by providing style images to a style signal generator CNN instead of using one-hot style conditional vectors. We train the network in an end-to-end manner with fifty thousand content images and fifty thousand style images (we used WikiArt images). One mini-batch contains the combination of one content image and several randomly-selected style images. The number of style images in one mini-batch depends on the amount of GPU memory.

4 Experimental Results

Figure 4 shows the results of a conditional style transfer network trained with 14 styles. Figure 5 shows the results of unseen style transfer by the proposed arbitrary style transfer network trained with 50,000 styles excluding all the styles shown in the figure. In the Appendix, we show the results including ones by Johnson et al. (2016) for qualitative comparison.

Additional results, demo videos and the iOS app of real-time multi-style transfer is available at http://foodcam.mobi/deepstylecam/ .

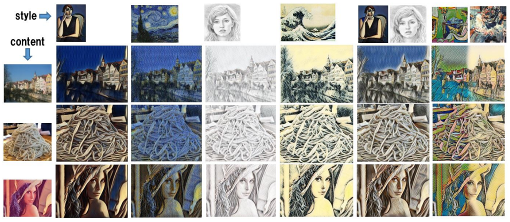

Figure 4: Results of a conditional transfer network. A mixed style and a spatial mixed style in the rightmost. All the styles are trained.

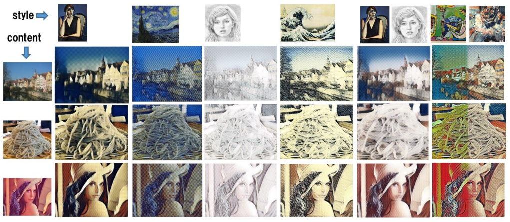

Figure 5: Results of unseen style transfer by the second network. All the styles shown above are *not* trained.

References

T. Q. Chen and M. Schmidr. Fast patch-based style transfer of arbitary style. In *arXiv:1612.04337*, 2016.

V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. In *arXiv:1610.07629*, 2016.

L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. In *arXiv:1508.06576*, 2015.

L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2016.

S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2016)*, 35(4), 2016.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. of International Conference on Machine Learning*, pp. 448–456, 2015.

J. Johnson, A. Alahi, and L. F. Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. of European Conference on Computer Vision*, 2016.

Y. Li, N. Wang, J. Liu, and X. Hou. Demystifying neural style transfer. In *arXiv:1701.01036*, 2017.

K. Simonyan, A. Vedaldi, and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Internation Conference on Machine Learning*, 2016a.

D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. In *arXiv:1607.08022*, 2016b.

Appendix

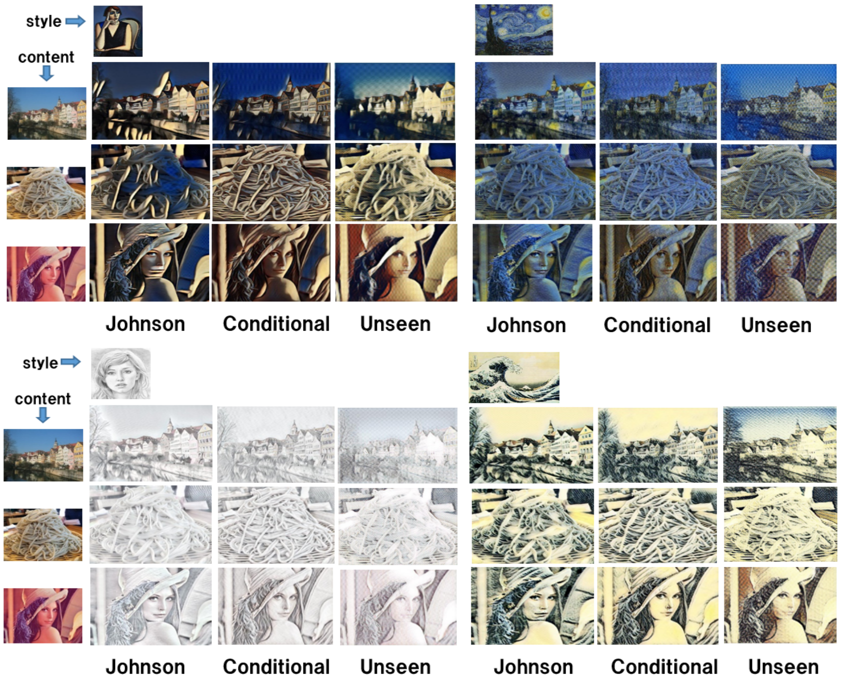

Figure 6: Qualitative comparison of the results by Johnson et al. (2016), Conditional Style Transfer (CST), and Unseen Style Transfer (UST). Note that the style images shown above were used for training in Johnson et al. (2016) and CST, while the above style images were NOT used for training in UST. The quality of Johnson's and CST are almost the same, while the results by UST is slightly different from them. However, we think UST is a good approximation of Johnson's and CST.