# Word-Conditioned Image Style Transfer

Yu Sugiyama and Keiji Yanai

Department of Informatics, The University of Electro-Communications, Tokyo
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585 JAPAN
{sugiya-y,yanai}@mm.inf.uec.ac.jp

**Abstract.** In recent years, deep learning has attracted attention not only as a method on image recognition but also as a technique for image generation and transformation. Above all, a method called Style Transfer is drawing much attention which can integrate two photos into one integrated photo regarding their content and style. Although many extended works including Fast Style Transfer have been proposed so far, all the extended methods including original one require a style image to modify the style of an input image. In this paper, we propose to use words expressing photo styles instead of using style images for neural image style transfer. In our method, we take into account the content of an input image to be stylized to decide a style for style transfer in addition to a given word. We implemented the propose method by modifying the network for arbitrary neural artistic stylization. By the experiments, we show that the proposed method has ability to change the style of an input image taking account of both a given word.

## 1 Introduction

In recent years, deep learning has attracted attention not only as a method on image recognition but also as a technique for image generation and transformation. Above all, a technology called Neural Style Transfer proposed by Gatys et al. [3] is drawing much attention as a method to synthesize an image which has the style of a given style image and the content of a given content image using a Convolutional Neural Network (CNN). However, the original method takes relatively longer time (typically several minutes) for stylizing images. Thus, many extended methods of Neural Style Transfer for fast stylization have been proposed so far. The most representative one is Fast Style Transfer proposed by Johnson et al. [7] which employs an encoder-decoder network with residual blocks for real-time image transfer instead of an optimization-based method of the original one. However, all the extended methods including both original one and fast one require to prepare a style image for image stylization.

For practical point of view, preparing style images are not always easy for every user, especially on the setting of using for smart-phone applications. A user usually have to selects a style from limited numbers of the images pre-registered in the system. Instead, words can be easily provided to the system, even when using smartphone applications, via on-screen keyboards or voice recognition.

Then, in this paper, we propose to use words representing photo styles instead of using style images for neural image style transfer. In our method, we take into account the content of an input image to decide a transferred style in addition to a given word. We implemented the propose method by modifying the network for arbitrary neural artistic stylization [4] which enabled real-time arbitrary style transfer. We added a new sub-network which generates a style conditional vector from a given word. To train a sub-network, we used adversarial training instead of using a standard L2 loss. By the experiments, we show that the proposed method has ability to change the style of an input image according to given words.

Note that the combination of text-based style image search and an arbitrary style transfer network can possibly achieve the same objective of ours. The advantage of our method over it is expected that a mixed style can be transferred automatically when an unknown style word which is not included in the training dataset is given.

## 2 Related Work

Many works on neural style transfer have been proposed so far. In this section, we explain the original method [3] and the fast-version of neural style transfer [7]. In addition, we describe some multiple style methods and fast arbitrary style methods including arbitrary neural artistic stylization [4] which is a base method we extend in this paper.

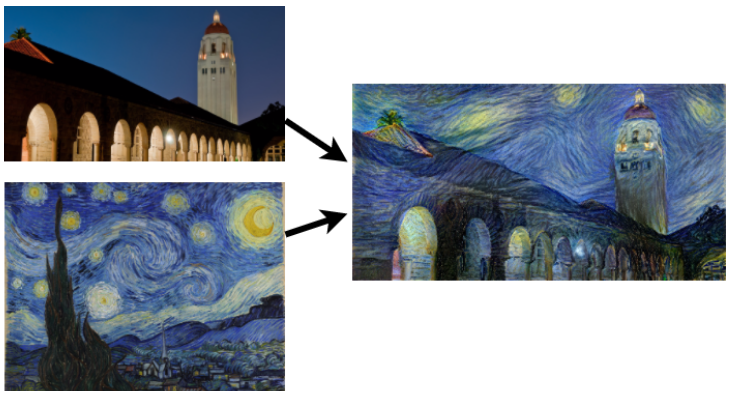

**Fig. 1.** An example of style transfer which integrates the content of a content image and the style of a style image.

### 2.1 Neural Style Transfer

Neural Image Style Transfer using Convolutional Neural Networks (CNN) proposed by Gatys et al. [3] is the first method of CNN-based image style transfer. In this method, given a style image and a content image, they generate a stylized image by optimizing an output image by minimizing the loss functions regarding both CNN activations of a content image and Gram matrix of a style image. It generates a stylized image by keeping the content of a content image and the style of a style image. For example, by integrating the content of a night church photo and the style of Gogh's starry night, we obtain a church painting in the style of Gogh as shown in Figure 1. This method enables us to modify the style of an image keeping the content of the image easily. It replaces the information which are degraded while the signal of the content image goes forward through the CNN layers with style information extracted from the style image, and reconstructs a new image which has the same content as a given content images and the same style as a given style image.

In this method, they introduced "style matrix" which was presented by Gram matrix of CNN activations, that is, correlation matrix between feature maps in CNN. Recently, it is indicated that a style feature which represents the style of an image is not only a Gram matrix of feature maps but also various kinds of statistics which represents the distributions of feature maps such as a combination of a mean and variance of each element of feature maps [8]. Since the original method proposed by Gatys et al. employs an optimization-based method for image generation, and requires both forward and backward computation iteratively to synthesize a stylized image, the processing time tends to be longer (several minutes) even using a GPU.

In this method, they optimized the sum of the content loss function (Eq.1) and the style loss function (Eq.4) by iteratively updating an output image. They calculated losses with VGG19 [10] without FC layers as a CNN feature extractor. The content loss function is shown below. $\boldsymbol{x_c}$ and $\boldsymbol{x_o}$ represents CNN feature vectors of an original input image and a generated image, and $F^l(\boldsymbol{x})$ are the feature representation of $\boldsymbol{x}$ in Layer $l$ of the VGG network.

$$L_{content}(\boldsymbol{x_c}, \boldsymbol{x_o}, l) = \frac{1}{2}||F^l(\boldsymbol{x_c}) - F^l(\boldsymbol{x_o})||^2 \tag{1}$$

We define a Gram matrix of a CNN feature vector, $F^l$, as $G^l$ which represents the style of an image. The style loss function is the sum of L2 losses between a gram matrix of a given style image and a gram matrix of the generated image over multiple layers.

$$G^l(\boldsymbol{x}) = (F^l(\boldsymbol{x}))(F^l(\boldsymbol{x}))^T \tag{2}$$

$$E_l(\boldsymbol{x_s}, \boldsymbol{x_o}) = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G^l(\boldsymbol{x_s}) - G^l(\boldsymbol{x_o}))^2 \tag{3}$$

$$L_{style}(\boldsymbol{x_s}, \boldsymbol{x_o}) = \sum_{l=0}^{L} E_l(\boldsymbol{x_s}, \boldsymbol{x_o}) \tag{4}$$

$$L_{total}(\boldsymbol{x_c}, \boldsymbol{x_s}, \boldsymbol{x_o}) = \alpha L_{content}(\boldsymbol{x_c}, \boldsymbol{x_o}) + \beta L_{style}(\boldsymbol{x_s}, \boldsymbol{x_o}) \tag{5}$$

It requires long time to optimize the pair of content image and style image. It is impossible to generate an image in instantly. The solution of this problem is Fast Style Transfer which employs an encoder-decoder network.

### 2.2 Fast Style Transfer

Fast Neural Style Transfer [7] achieved an instant style transfer with a pre-trained encoder-decoder network which enables real-time style transfer by one-time feed-forward computation. Basically the loss functions are the same as optimization-based method by Gatys et al. [3]. In fast style transfer, an encoder-decoder network is optimized instead of optimizing output images.

To accelerate neural style transfer, 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 [7, 13].

Johnson et al. proposed a perceptual loss to train a encoder-decoder network as a feed-forward style transfer network [7]. Their network can generate a stylized image for a given content image regarding consists of down-sampling layers, convolutional layers and up-sampling layers, which accepts an content image and outputs an synthesized image integrated with a fixed pre-trained style in real-time.

### 2.3 Multiple Style Transfer

Although Johnson et al.'s feed-forward network can treat only one fixed style, recently Dumoulin et al. [2] proposed a method to learn multiple styles with an encoder-decoder fast style transfer network. They used conditional version of Instance Normalization [14] instead of Batch Normalization [6] 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.

The other approach on multiple style transfer is "Conditional Fast Style Neural Network" proposed by Yanai [15]. The idea on introducing conditions

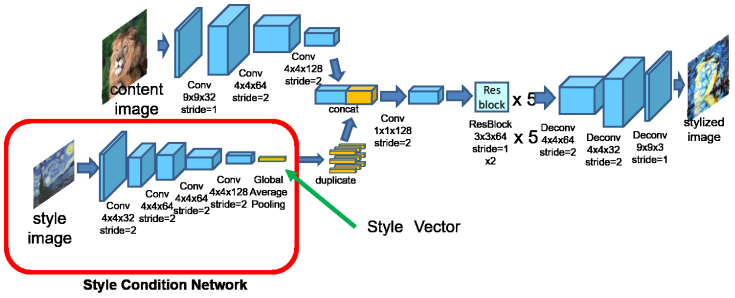


**Fig. 2.** Unseen Style Transfer Network [15].

for style selection is similar to Demoulin et al. Dumoulin et al. introduced a new special layer, "conditional instance normalization" layer, while Yanai added an additional input signal which is concatenated with an internal activation signal and we introduced one additional $1 \times 1$ convolution layer to integrate an internal signal and a style signal. They use only common layers in the proposed networks. In addition, in their method, to mix multiple styles, they just assign weight values to the multiple elements of the conditional input vector such as $(0.2, 0.3, 0.1, 0.4)$.

### 2.4 Unseen/Arbitrary Style Transfer

More recently, a fast arbitrary style transfer method which can transfer even untrained styles has been proposed by Chen et al. [1]. 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. [3]. However, it takes more than one second to generate a stylized image, since style swapping is a little bit complicated processing.

Yanai [15] propose a feed-forward network for arbitrary style transfer by extending "conditional fast style transfer network" (Figure 2). In his 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, he 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.

They 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. They trained the conditional style transfer network with a style condition generator network in an end-to-end manner, and showed that it worked as arbitrary style network. The basic idea for arbitrary

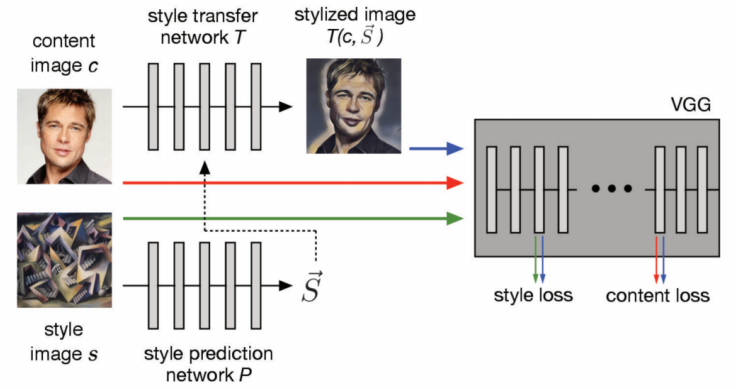


**Fig. 3.** Arbitrary neural artistic stylization network [4].

style transfer is different from [1]. The 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.

Ghiasi et al. proposed an arbitrary neural artistic stylization network [4] (Figure 3). The idea of this network is similar to [15]. The difference is that this network generates parameters of instance normalization layers directly by a style condition generator network. This can be regarded as an extension of the multiple style network employing conditional instance normalization [2]. The quality of stylization is superior to "Unseen Style Network" [15] since it generates all the instance normalization parameters depending on given style images instead of simple concatenation of conditional vectors.

In this paper, we use an arbitrary neural artistic stylization network [4] as a base network, and extend it so as to achieve word-based style transfer. To say it concretely, we replace the part of a Style Prediction Network (SPN) in the network with a Style Selector Network (SSN) which generates a conditional signal from a given word. Note that we made pre-liminary experiments with "Unseen Style Transfer Network" [15] as well. It turned out that the arbitrary neural artistic stylization network [4] was more appropriate for our purpose.

## 3 Proposed Method

### 3.1 Overview

In the proposed method, we replace the part of a Style Prediction Network (SPN) in the arbitrary neural artistic stylization network [4] with a newly-introduced Word-based Style Selector Network (SSN) as shown in Figure 4.

We use the arbitrary stylization network pre-trained with various kinds of content images and style images according to [4] as a base network. The arbitrary stylization network consists two networks: The main network is a style transfer

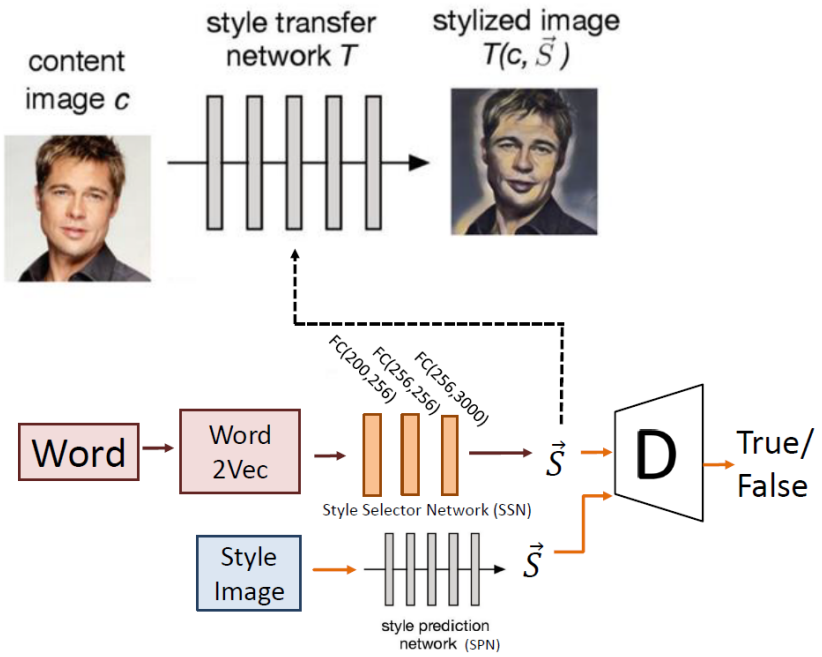


**Fig. 4.** The arbitrary stylization network with Style Selector Network (SSN) with adversarial training. In the figure, D represents the discriminator which classifies style vectors generated from SSN from style vectors generated from SPN.

network which is based on Conv-Deconv network with Residual Block proposed by Johnson et al. [7]. Note that all the batch normalization layers in the Johnson-style network were replaced with conditional instance normalization layers. The additional one is a style prediction network (SPN) which generates parameters of instance normalization layers. Conditional instance normalization normalizes each unit's activation $z$ as

$$z = \gamma_s \frac{z - \mu}{\sigma} + \beta_s \tag{6}$$

where $\mu$ and $\sigma$ are the mean and standard deviation across the spatial axes in an activation map [2]. $\gamma_s$ and $\beta_s$ constitute a linear transformation that specify the learned mean ($\beta_s$) and learned standard deviation ($\gamma_s$) of the unit. This linear transformation is unique to each style $s$. In particular, the concatenation $\tilde{S} = \gamma_s, \beta_s$ constitutes a roughly 3000-d embedding vector, a style vector, representing the style. SPN consists of a pre-trained Inception-v3 [11] and two additional fully-connected layers.

In the proposed network, we introduce Word-based Style Selector Network (SSN). We train the SSN using word-annotated style training images so that the output signal of the SSN approximates the output of the pre-trained SPN. To make the distribution of the output of SSN closer to the one of the output of SPN, we use adversarial training [5] instead of standard L2 minimization. An

input of the SSN is a word embedding vector of a given word represented by Word2Vec, and the output is a style vector the dimension of which is roughly 3000.

### 3.2 Word Embedding

In the proposed method, we need to convert a word to a vector to provide it into the Style Selector Network. To do that, we use Word2Vec [9]. In the experiments, we used Japanese adjective words for representing Photo Style. Therefore we trained the Word2Vec model using open Japanese WikiPedia data containing many Japanese adjective expressions after pre-processing including removing low-frequency words and non-independent words which are neither nouns, adjectives nor verbs. We convert a given input word into a 200-dim word embedding vector with the trained Word2Vec model.

### 3.3 Style Selector Network

The Style Selector Network (SSN) (Figure 4) generates a style vector corresponding to a given word expressing an image style instead of the Style Prediction Network (Figure 3) which generates a style vector from a style input image in the original arbitrary stylization network.

The SSN takes a 200-d work embedding vector of a given word as an input. SSN consists of three fully-connected (FC) layers in Figure 4. Note that each of the FC layers has a ReLU activation function and a Batch Normalization layer after it.

For training the SSN, we use the trained Style Prediction Network (SPN) which is originally a part of the arbitrary stylization network to generate ground-truth style vectors of training samples as shown in Figure 4. For all the training word-annotated style images, we extract style vectors, $s_i$, using the trained SPN, and we obtain a 200-d word embedding vector, $w_i$, by Word2Vec. An input of the SSN is a word embedding vector of a given word represented by Word2Vec, and the output is a style vector the dimension of which is roughly 3000.

Although we can train the SSN by minimize a L2 loss function, to make the distribution of the output of SSN closer to the one of the output of SPN, we use adversarial training [5]. For adversarial training, we prepare a discriminator, $D$, which classifies style vectors generated from SSN from style vectors generated from SPN. SSN and $D$ are trained alternatively so that $D$ can discriminate them correctly and SSN can generate word-based style vectors which $D$ classifies as image-based style vectors SPN generated. $D$ consists of four FC layers each of which has batch normalization layers and ReLU after it except the last layer. The adversarial loss function is as follows:

$$L_{adv} = \mathbb{E}_{x \sim P_{data}(x)}[\log D(SPN(x))] + \mathbb{E}_{w \sim P_{data}(w)}[\log(1 - D(SSN(w)))] \quad (7)$$

where $x$ and $w$ represent a style image and a style word, respectively. We minimize this loss function for training of $SSN$, while we maximize this for training of

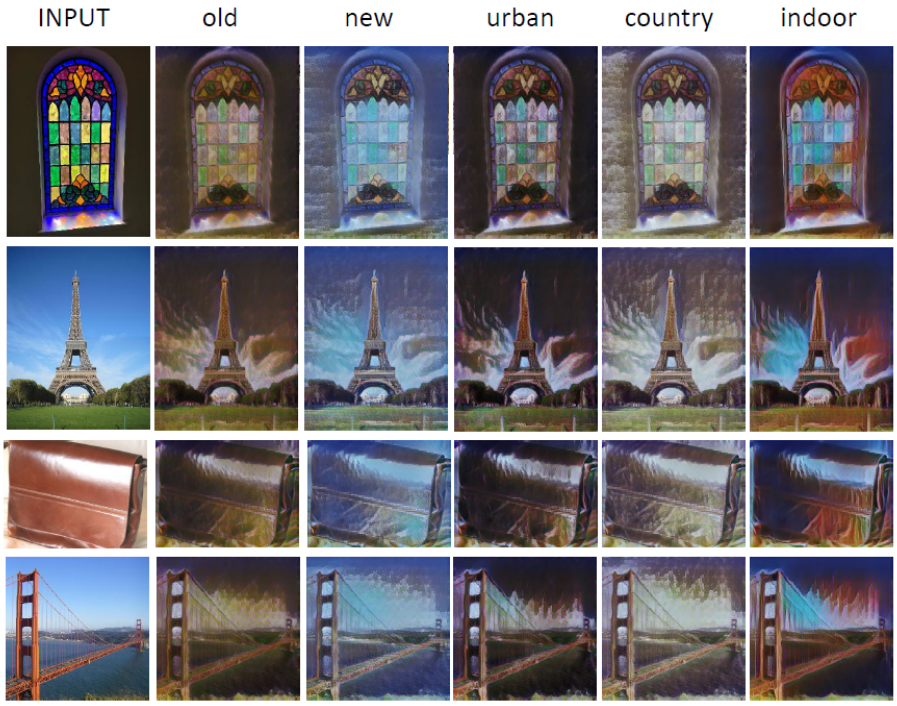


**Fig. 5.** The word-based stylized results with adversarial training.

the discriminator, $D$, i.e. $\min_{SSN} \max_D L_{adv}$. Note that $SPN$ is the pre-trained model and kept fixed while training.

At the time of transformation of images, we remove SPN and the discriminator, and insert the SSN instead. We use the whole network as a word-based style transfer network which takes a word and an image as inputs.

## 4 Experiments

To train the Style Selector Network, we need to prepare style images with words expressing their styles. We select 500,000 images annotated with adjective words randomly from the Yahoo Flickr Creative Commons 100 Million Dataset (YFCC100M) [12] which are made of Creative Commoms images in the Flickr database. Note that we used the official pre-trained model of the arbitrary stylization network including the trained parameters of both a style transfer network and a style prediction network (SPN) available from the GitHub [1].

We made experiments with the trained network. Figure 5 shows the results by the model trained with 500,000 YFCC100M images for five words, "old",

[1] https://github.com/tensorflow/magenta/


**Fig. 6.** The word-based stylized results without adversarial training using L2 loss minimization.

"new", "urban", "country" and "indoor". The styles were changed depending on style words, and all the images were stylized in the same way for the same word. In fact, we generated style vectors directly from only words. Therefore, the correspondence between a word and a style vector is one-to-one. To make results more diverse, one of the possible solutions is introducing small randomness by adding a random variable like a standard GAN for image generation. Among opposite meaning word pairs such as "old" and "new", their styles were largely different. However, both styles were relatively dark styles. The possible reason of this might be that mixing many styles or colors brings darker styles and colors in general.

Figure 6 shows the results with the model trained with L2 loss instead of adversarial loss. The differences of the styles among different words were very small, which indicated that the training of SSN did not succeed. Adversarial training was more effective for training of SSN than the standard L2 minimization.

As additional experiments, we trained the proposed model with Web image data we gathered by ourselves. We limited adjective words to ones related to "leather" for query words for Web image search. We picked up 84 highly-frequent adjective words with "leather" using Web search engines, and gathered 32,898 images with the combined query words of "leather" and one of the 84 adjective words using Web image search engines. Figure 7 shows the results generated by the model trained with word-associated Web images of only leather-related adjective words. Since "elderly", "fossil" and "old" were synonyms, the stylized images tends to be similar each other. On the other hand, among other three words the results are different.


**Fig. 7.** The word-based stylized results with adjective-associated images gathered from the Web.

## 5 Conclusions

In this paper, we proposed the fast style transfer network which modify the style of a given image based on a given word expressing image styles. By the proposed method, we have achieved word-based image style transfer which accepts any words as an input. In the experiments, we confirmed that different words generate different stylized images. In addition, the results with adversarial training seems to be superior to ones with L2 loss.

The results shown in this paper was still preliminary, and we just confirmed the proposed method worked. Comprehensive evaluation including evaluation by subjects is needed for future work. In addition, because in the current network one word corresponds to one style and the diversity was limited, we plan to introduce randomness by adding random variable to the input of SSN, or take into account the content of given images by adding image feature vectors of the given images to the input of SSN as additional information.

In this work, we extended a style transfer method which keeps content vectors of a content image and style vectors of a style image represented by Gram matrix. As results, the obtained results look like the images stylized by the average of many images associated with given adjective words. For future work, we plan to introduce more generic image transformation methods such as CycleGAN [16] for word-based image transformation.

**Acknowledgments:** This work was supported by JSPS KAKENHI Grant Number 15H05915, 17H01745, 17H05972, 17H06026 and 17H06100.

## References

1. Chen, T.Q., Schmidt, M.: Fast patch-based style transfer of arbitrary style. arXiv preprint arXiv:1612.04337 (2016)

2. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. Proc. of ICLR (2017)
3. Gatys, L.A., Ecker, A.S., Bethge., M.: Image style transfer using convolutional neural networks. In: Proc.of IEEE Computer Vision and Pattern Recognition. pp. 2414–2423 (2016)
4. Ghiasi, G., Lee, H., Kudlur, M., Dumoulin, V., Shlens, J.: Exploring the structure of a real-time, arbitrary neural artistic stylization network. arXiv preprint arXiv:1705.06830 (2017)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proc.of Advances in Neural Information Processing Systems 25. pp. 2672–2680 (2014)
6. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
7. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Proc.of European Conference on Computer Vision. pp. 694–711 (2016)
8. Li, Y., Wang, N., Liu, J., Hou, X.: Demystifying neural style transfer. arXiv preprint arXiv:1701.01036 (2017)
9. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean., J.: Distributed representations of words and phrases and their compositionality. In: Proc.of Advances in Neural Information Processing Systems 25. pp. 3111–3119 (2013)
10. Simonyan, K., Vedaldi, A., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proc.of International Conference on Learning Representation (2015)
11. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
12. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. Communications of the ACM **59**(2), 64–73
13. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: Feed-forward synthesis of textures and stylized images. In: ICML. pp. 1349–1357 (2016)
14. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization (2016)
15. Yanai, K.: Unseen style transfer based on a conditional fast style transfer network. In: Proc.of International Conference on Learning Representation Workshop Track (ICLR WS) (2017)
16. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision (2017)