
DEPTH-PRESERVING REAL-TIME ARBITRARY STYLE TRANSFER

Konstantin Kozlovsev

Lomonosov Moscow State University
kozlovsev179@gmail.com

Victor Kitov

Lomonosov Moscow State University,
Plekhanov Russian University of Economics
v.v.kitov@yandex.ru

June 5, 2019

ABSTRACT

Style transfer is the process of rendering one image with some content in the style of another image, representing the style. Recent studies of Liu *et al.* (2017) [1] have shown significant improvement of style transfer rendering quality by adjusting traditional methods [2],[3] with regularizer, forcing preservation of the depth map of the content image. However these traditional methods are either computationally inefficient or require training a separate neural network for new style. AdaIN method of Huang *et al.* (2017) [4] allows efficient transferring of arbitrary style without training a separate model but is not able to reproduce the depth map of the content image. We propose an extension to this method, allowing depth map preservation. Qualitative analysis and results of user evaluation study indicate that the proposed method provides better stylizations, compared to the original style transfer methods [2] and [4].

Keywords image processing · image generation · depth estimation · instance normalization

1 Introduction

The problem of rendering an image (called the *content image*) in a particular style is known as *style transfer* and is a long studied problem in computer vision. Early approaches [5, 6, 7] used algorithms with human engineered features targeting to impose particular styles.

In 2016 Gatys *et al.* [2] proposed an algorithm of imposing arbitrary style taken from user defined *style image* on arbitrary content image by using representations of images that could be obtained with deep convolutional networks. However their method needed a computationally expensive optimization in the space of images requiring several minutes of processing a single image of moderate resolution on powerful GPUs. In later works Ulyanov *et al.* [8] and Jonson *et al.* [3] proposed to use an end-to-end convolutional network for image transformation. In their method, the network was trained to transform any input image to the given style, according to the loss function used by Gatys *et al.* [2]. These methods worked fast, but required training a separate transformation network for a new style. Work of Liu et. al (2017) [1] extended traditional methods [2],[3] with a regularizer, forcing preservation of the depth map of the content image. This yielded significant improvement of style transfer rendering quality since depth was essential in human perception and analysis of images. Later architectures, such as AdaIN [4] and other ([9], [10]), allowed transferring arbitrary style without training a separate network but lacked rendering quality due to failure to preserve the depth map of the content image.

In this work AdaIN method is extended to allow preservation of the depth map. Specific structure of the method does not allow adding depth regularizer, as in [1]. The main idea of this method is to stylize image according to its perceptual depth, that could be found with using contemporary depth estimation models, such as of Chen *et al.* [11]. We suggest that closer objects on the image should be less stylized than distant ones, and use spatial style strength control to stylize different parts of image with different strength. The main advance

of our method is that it can be used in real-time for any pair of content and style images. We demonstrate that the proposed extension improves the quality of the original AdaIn method and provides stylizations of comparable quality to [2], being much more efficient.

The remainder of paper organized as follows. Section 2 gives an overview of the related methods. Section 3 describes proposed method and specific details of approaches that are related to it. In section 4 we provide experimental results and compare our method with alternative style transfer methods qualitatively and with the help of a user evaluation study. Finally section 5 concludes.

2 Related work

Image stylization is a long studied problem in computer vision. It takes its origin from non-photorealistic rendering [12] and texture generation [13, 14, 15] tasks. Early approaches [5, 6, 7] relied on low-level hand-crafted features and often failed to capture semantic structures. Later work of Gatys *et al.* [2] proposed a new style transfer algorithm that was flexible enough to stylize any content image using arbitrary style extracted from any *style image* and showed impressive results. It performed stylization by matching Gram-matrices of image features taken from layers of a deep convolutional classification network VGG-19 [16].

However method of Gatys *et al.* is very computationally demanding. The generation of single image with moderate resolution takes several minutes even on modern GPUs. So later works by Ulyanov [8] and Jonson [3] proposed to impose style by passing a content image through a transformation network, trained to impose fixed style on a big dataset of content images. The transformation network is trained to solve the optimization task introduced in Gatys *et al.*. Thus trained model is tied to one specific style, and imposing another style requires a separate model.

In later works, Dumoulin *et al.* [17] and Chen *et al.* [18] proposed to use so-called *style banks* for different styles to reduce the number of parameters for multi-style rendering. The main idea was to let only a small portion of the network to control the style, all other parameters were held constant for different styles. Chen *et al.* proposed to use convolutional layers in the middle of end-to-end transform network as style banks, and Dumoulin *et al.* used parameters of *instance normalization layers* [8] for style representations. This approach allowed to use several different styles for fast style transfer, but still it required to train some subset of parameters for each new style. In [9] it was proposed to use separate model, predicting parameters of instance normalization, given the style image, rather than train them.

Alternative approach was used in AdaIN style transfer method [4]. The main idea of the paper was to replace instance normalization layers with *adaptive instance normalization* (AdaIN), which first normalized content features to have zero means and unit standard deviations, and then rescaled them with means and standard deviations obtained from the style image representation. The AdaIN method has the advantage of fast stylization by just passing the content image through the transformation network. Also it can be applied to any style and does not require training, because only means of standard deviations of style image inner representation are necessary for transformation. Since style has straightforward representation, it becomes possible to control global stylization strength and mix different styles.

The other arbitrary real time method was proposed by Li *et al.* [10], so-called *universal* style transfer. This approach considered pretrained deep convolution encoder (they also used VGG-19 architecture) for high-features extraction and decoders, which can reconstruct images from hidden representation from different layers of VGG. For style transfer they applied whitening and linear transformation which imposed the style, represented as mean vector and covariance matrix of the transformation. Stylization results were obtained after passing hidden representations through the trained decoders.

The task of depth-preserving style transfer was first covered by Liu *et al.* [1], in their work the transformation network from Jonson’s paper [3] learned to generate images that were also close to content by its depth, where image depth was calculated with network proposed by Chen *et al.* [11].

Style can also be imposed using generative adversarial networks (GANs). For example, Zhu *et al.* [19] considered cycle GANs for transferring images between two different domains, Zhang *et al.* [20] used GANs to colorize sketches. Difference, compared to style transfer, is that GANs require many style images to reconstruct the style.

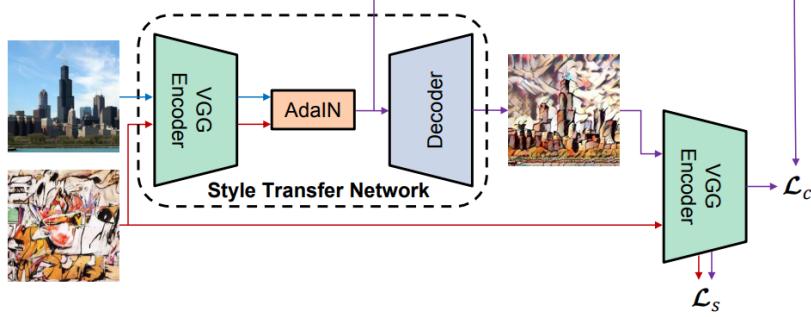


Figure 1: AdaIN style transfer architecture [4].

3 Methods

3.1 AdaIN method

In AdaIN method [4] any content image I_c can be stylized using any style image I_s . Stylization result \hat{I} is obtained using

$$\hat{I} = g(\text{AdaIN}(f(I_c), f(I_s))).$$

where $f(\cdot)$, $g(\cdot)$ are encoder and decoder respectively, and $\text{AdaIN}(x, y)$ is a variant of instance normalization [8], where instance normalization parameters are taken from the style image representation. Define $x = f(I_c)$, $y = f(I_s)$, $x \in \mathbb{R}^{C \times H \times W}$ and style image $y \in \mathbb{R}^{C \times H \times W}$. Then $\text{AdaIN}(x, y) \in \mathbb{R}^{C \times H \times W}$ and is defined as

$$\text{AdaIN}(x, y)_{cij} = \sigma_c(y) \left(\frac{x_{cij} - \mu_c(x)}{\sigma_c(x)} \right) + \mu_c(y), \quad c = 1, 2, \dots, C. \quad (1)$$

$$\mu_c(x) = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W x_{cij}, \quad \sigma_c(x) = \sqrt{\frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W (x_{cij} - \mu_c(x))^2} \quad (2)$$

For style transfer the simple encoder-decoder transform network is used (fig. 1). The encoder is a fully-convolutional part (until relu4_1 layer) of VGG network, pre-trained on ImageNet dataset [21] (and its weights are fixed in the following decoder training). Decoder $g(x)$ is symmetric to the encoder: every max-pooling operator is replaced with $\times 2$ nearest neighbors upsample layer, and all convolution-ReLu blocks are replaced with the same block, except the last one, which has no ReLu layer after it. So the output of decoder has the same dimensionality as the content image I_c .

One of the advantages of this approaches is that, after learning a decoder, we can use AdaIN outputs as a hidden representation of the stylized image. By mixing it with content image representation we can control stylization strength.

3.2 Depth estimation

Many image depth estimation models exist. We use proposed by Chen *et al.* [11] end-to-end network, which makes predictions using only the image itself. This method is trained for relative depth reconstruction, so its output can only distinguish which pixel on image is closer or more distant (by assigning lesser or larger values to the pixel). For our method we rescale estimated image depth map to $[0, 1]$ segment, by subtracting minimal value and dividing by the difference between maximal and minimal value. Thus 0 corresponds to most close pixels and 1 - to most distant ones. It is possible to adjust minimal and maximum values to control impact of depth on the strength of the imposed style.

3.3 Proposed extension

Proposed method extends AdaIN method and is specifically targeted to preserve depth information of the content image. This is done to improve the quality of stylizations, since depth is essential for human perception of images. Since stylization partially distorts the content image, we propose to apply stronger stylization to

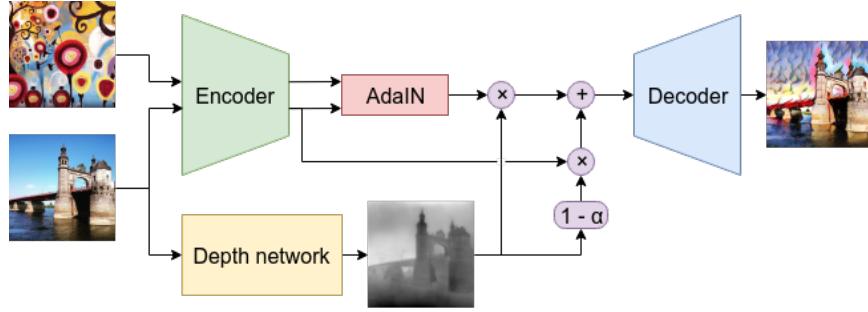


Figure 2: Architecture of the proposed method with style strength control.

closer objects and weaker stylization to more distant objects on the content image. We name our method *AdaIN-DCS* (adaptive instance normalization-depth controlled strength), since our approach preserves depth information by spatially adjusting the strength of imposed style.

Let $f(I_c)$ correspond to the non-stylized content representation and $\text{AdaIN}(f(I_c), f(I_s))$ - to stylized one. One can control global stylization strength, using

$$\hat{I} = g(\alpha \text{AdaIN}(f(I_c), f(I_s)) + (1 - \alpha)f(I_c))$$

Since we are interested in stylization with spatially varying strength, from 3.2 we obtain a depth map $D \in \mathbb{R}^{H \times W}$ for style strength control. Stylization is performed, using

$$\hat{I} = g(D \odot \text{AdaIN}(f(I_c), f(I_s)) + (1 - D) \odot f(I_c)),$$

where \odot denotes element-wise multiplication. This architecture is illustrated on fig. 2.

For stylization pre-trained AdaIN encoder/decoder [16, 4] and pre-trained depth network [11] is used. Computational advantage of our method is that it is learning-free: given pretrained encoder, decoder and depth estimation network, method does not require additional training for new styles.

4 Experiments

4.1 Style strength control

Firstly we demonstrate how proposed spatial style strength control works on an illustrative example. Consider content, style and linear gradient mask and stylization result on fig. 3. Since depth map increases as we move left to right, stylization strength gradually increases in the same direction, according to the mask values.

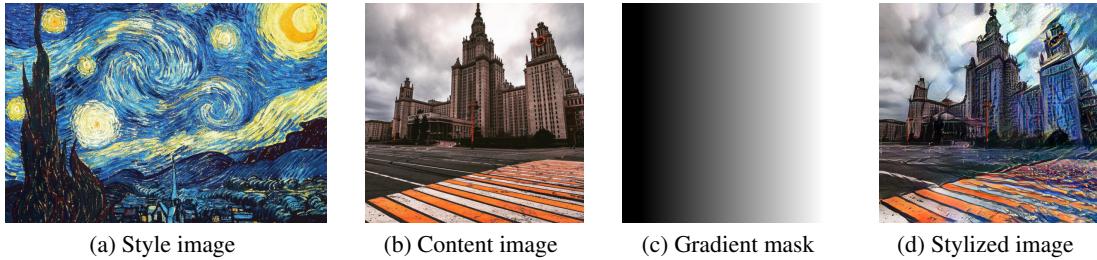


Figure 3: Demonstration of the spatial style strength control with a linear gradient mask. Stylization strength smoothly increases as the mask values vary from 0 to 1.

This experiment illustrates that it is possible to combine stylized image with non-stylized not only using the discrete 0/1 hard mask, as shown in [4] for different styles, but also with any real values inside the $[0, 1]$ interval.

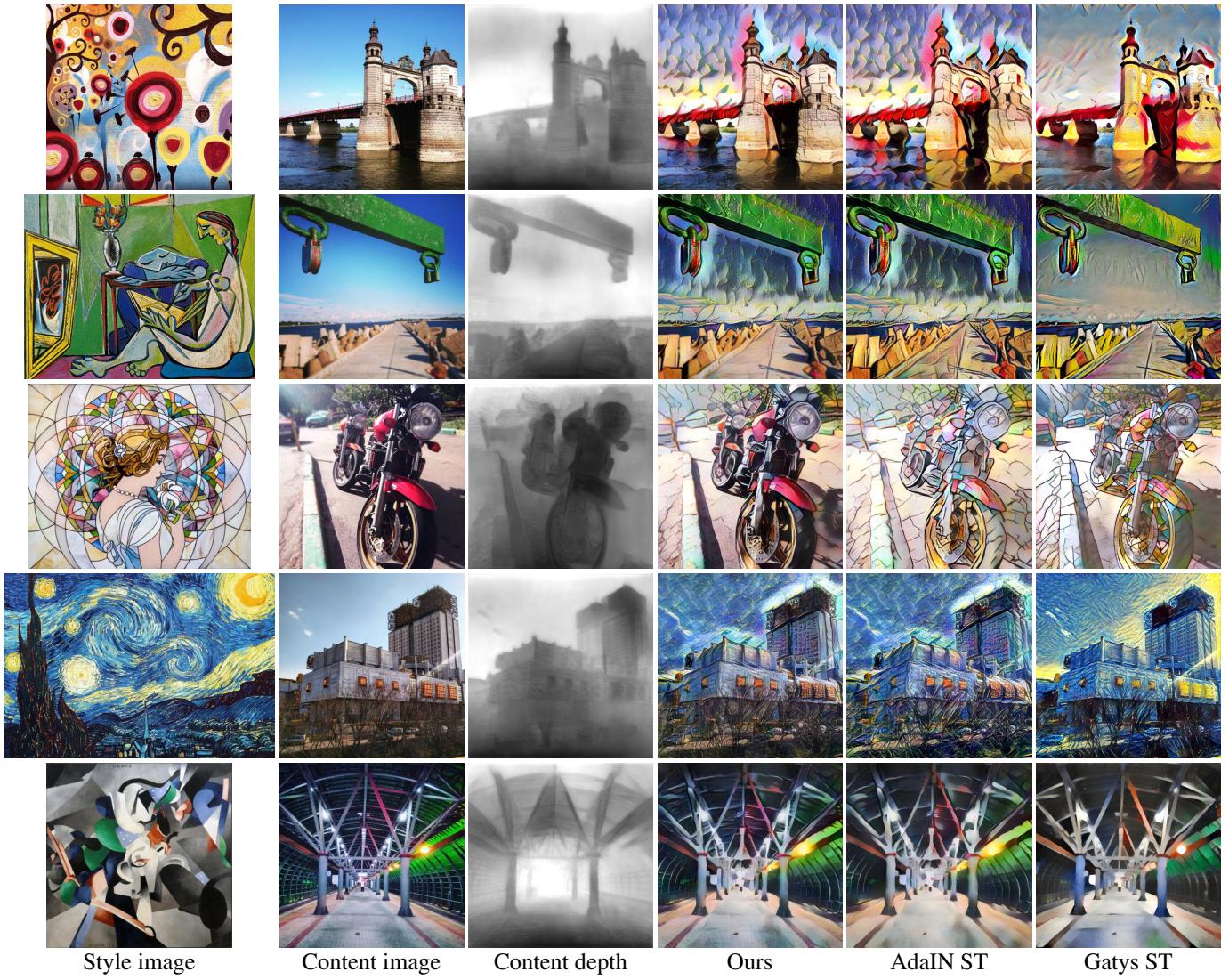


Figure 4: Our method vs AdaIN and Gatys methods.

4.2 Qualitative comparison

We provide a qualitative comparison of our method with the classical optimization-based style transfer by Gatys *et al.* and the AdaIN method. Stylization results of the compared methods are shown on fig. 4 for images, randomly picked from <https://instagram.com>. It can be observed that depth network adequately estimates depth for the considered images, AdaIN and optimization-based style transfer methods stylize whole image uniformly with the same style strength, and the proposed method applies style less to more close objects, rendering them in a more detailed way, which produces visually pleasing result.

Comparisons of properties of major style transfer methods can be seen on table 1. We consider the followed features:

Real-time: the algorithm is considered real-time if it can generate stylized image in one forward pass through the neural network.

Number of styles: the number of styles that can be applied to content image during inference.

Learning-free: algorithm is learning-free if it requires no training for new styles.

Depth preservation: whether algorithm is capable to preserve the depth of the content image or not.

As we can see from table 1 our method possesses all of the considered advantages.

Method	Real-time	#Styles	Depth preservation	Learning-free
Gatys <i>et al.</i>	✗	∞	✗	✗
Jonson <i>et al.</i>	✓	1	✗	✗
Dumoulin <i>et al.</i>	✓	32	✗	✗
Ghiasi <i>et al.</i>	✓	∞	✗	✓
AdaIN (Huang and Belongie)	✓	∞	✗	✓
Liu <i>et al.</i>	✓	1	✓	✗
Universal (Li <i>et al.</i>)	✓	∞	✗	✓
Ours	✓	∞	✓	✓

Table 1: Features of major style transfer methods. Proposed method possesses all of the features.

4.3 User evaluation study

Procedure. To provide a more general comparison of style transfer methods we conduct a user evaluation study. The study consists of surveys and each survey allows to compare two stylization methods. In each survey a number of content images are stylized using a number of style images. For each content-style pair a respondent is shown a pair of stylizations with the two compared methods and he is asked to select the result he aesthetically likes more. To omit position bias, for each content-style pair stylizations by different methods appear in random order. We did not tell the respondents anything about the depth preservation concept and our algorithm details.

Setup. We conduct two surveys: our method vs. AdaIN and our method vs. optimization-based style transfer by Gatys *et al.* To omit image selection bias we take 6 typical style images, used in style transfer literature, and 7 content images. Content images were not cherry picked, the only limitation was that they should contain objects with different proximity to the viewer. Then all contents were stylized using all styles yielding 42 stylizations in total. No filtering was performed for the sake of fairness of the experiments. Stylization strengths of AdaIN and optimization-based style transfer were adapted to match average stylization strength of our method to make results comparable.

Results. The results of user study evaluation study are presented on table 2. Our method receives significantly higher evaluation than AdaIN, which validates the importance of preserving information about depth on the content image. Also our method receives higher score compared to optimization-based style transfer method of Gatys *et al.* Although margin is not big this time, our method performs stylization in real-time without computationally intensive optimization used by the method of Gatys *et al.*

Table 2: Results of user evaluation study

Experiment	Ours vs Gatys	Ours vs AdaIN
# image pairs	42	42
# respondents	10	16
# responses	420	672
# votes for proposed method	242	567
proportion of votes for proposed method	57.61%	84.37%

5 Conclusion

We propose an extension to AdaIN method, allowing to preserve depth information from the content image. All other benefits of AdaIN are preserved, namely fast real-time stylization and the ability to transfer arbitrary style at inference time without additionally training the model. Qualitative analysis reveals that the proposed method is capable to preserve information about proximity to objects on the stylized image and results of the user evaluation study highlight that depth preservation is important for the users, making them to prefer our method compared to AdaIN and optimization-based style transfer methods.

References

- [1] Xiao-Chang Liu, Ming-Ming Cheng, Yu-Kun Lai, and Paul L Rosin. Depth-aware neural style transfer. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, page 4. ACM, 2017.
- [2] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.
- [3] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.
- [4] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.
- [5] Bruce Gooch and Amy Gooch. *Non-photorealistic rendering*. AK Peters/CRC Press, 2001.
- [6] Thomas Strothotte and Stefan Schlechtweg. *Non-photorealistic computer graphics: modeling, rendering, and animation*. Morgan Kaufmann, 2002.
- [7] Paul Rosin and John Collomosse. *Image and video-based artistic stylisation*, volume 42. Springer Science & Business Media, 2012.
- [8] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.
- [9] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*, 2017.
- [10] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in neural information processing systems*, pages 386–396, 2017.
- [11] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016.
- [12] 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, 2012.
- [13] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999.
- [14] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346. ACM, 2001.
- [15] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in neural information processing systems*, pages 262–270, 2015.
- [16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [17] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *Proc. of ICLR*, 2, 2017.
- [18] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1897–1906, 2017.
- [19] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- [20] Lvmin Zhang, Yi Ji, Xin Lin, and Chunping Liu. Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 506–511. IEEE, 2017.
- [21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.