Real-Time Style Transfer with Strength Control

Victor Kitov[1,2(✉)]

[1] Lomonosov Moscow State University, Moscow, Russia
[2] Plekhanov Russian University of Economics, Moscow, Russia
v.v.kitov@yandex.ru
https://victorkitov.github.io

Abstract. *Style transfer* is a problem of rendering a content image in the style of another style image. A natural and common practical task in applications of style transfer is to adjust the strength of stylization. Algorithm of Gatys et al. [4] provides this ability by changing the weighting factors of content and style losses but is computationally inefficient. *Real-time style transfer* introduced by Johnson et al. [9] enables fast stylization of any image by passing it through a pre-trained transformer network. Although fast, this architecture is not able to continuously adjust style strength. We propose an extension to real-time style transfer that allows direct control of style strength at inference, still requiring only a single transformer network. We conduct qualitative and quantitative experiments that demonstrate that the proposed method is capable of smooth stylization strength control and removes certain stylization artifacts appearing in the original real-time style transfer method. Comparisons with alternative real-time style transfer algorithms, capable of adjusting stylization strength, show that our method reproduces style with more details.

Keywords: Image processing · Image generation · Style transfer · Texture synthesis · Stylization strength · Residual network · Multi-task learning

1 Introduction

Gatys et al. [4] demonstrated that deep neural networks can represent not only the content but also the style of the image which can be described by the Gramm matrices, containing covariances between activations at different channels of the deep convolutional network. The disentanglement of content and style enabled neural style transfer - a technique to render any content image in the style taken from another style image using deep neural network. Since content image already possesses some inherent style, it becomes necessary to specify the amount of style that needs to be transferred from the style image. Original approach of Gatys et al. [4] allowed to do that by adjusting the weights besides content and

M. Vento and G. Percannella (Eds.): CAIP 2019, LNCS 11679, pp. 206–218, 2019.
https://doi.org/10.1007/978-3-030-29891-3_19

style components in the target loss function. However, this approach required computationally expensive optimization process in the space of pixel intensities of the stylized image.

Later works of Ulyanov et al. [16] and Johnson et al. [9] (which we refer to as *real-time style transfer* or *the baseline method* for short), proposed a fast framework for style transfer. In this approach an image transformer network was trained and then any image could be stylized in real-time by passing it through this network. Since transformer was trained using loss function of Gatys et al. consisting of a weighted sum of style and content loss components, a change in desired stylization strength implied a change in the optimization criteria and thus required training a separate transformer network. This incurred not only computational costs for training multiple models, but also storage costs for keeping them on disk. More importantly, this approach suffered from the limitation that stylization with only a discrete set of stylization strengths could be applied - one for each trained transformer network, whereas stylization strength is inherently a continuous feature.

We propose a modification for the style transfer approach of Johnson et al. [9], which we name *real-time style transfer with strength control*. Proposed algorithm retains the advantage of the original method - namely, it applies style to any image very fast by just passing it through a feed-forward transformer network. However, it additionally gives the possibility to continuously adjust stylization strength at inference time.

The proposed architecture yields comparable quantitative results to the baseline algorithm, measured by the total loss value that is minimized for both methods. Qualitatively, for higher stylization strength it gives stylization of comparable quality to the baseline. Interestingly, for smaller stylization strength proposed method gives results of higher quality by alleviating stylization artifacts consistently generated by the baseline method. This observation is illustrated on sample images and further supported by the results of the user evaluation study.

Qualitative comparison is provided with another modern stylization methods, capable to control stylization strength at inference time. Namely, we compare our method with AdaIn [8] and universal style transfer [11]. Results demonstrate that our method reproduces style with significantly more details, while AdaIn and universal style transfer drop much of the style information.

The paper is organized as follows. Section 2 gives an overview of related methods. Section 2 describes the proposed method in detail. Section 4 provides experimental results comparing proposed algorithm with existing real-time style transfer method qualitatively, quantitatively and by means of a user study. It also provides qualitative comparison of our method with two other methods capable of performing stylization with strength control. Finally, Sect. 5 concludes.

2 Related Work

The task of rendering image in given style, also known as style transfer and non-photorealistic rendering, is a long studied problem in computer vision. Earlier approaches [6,13,15] mainly targeted reproduction of specific styles (such as

pencil drawings or oil paintings) and used hand-crafted features for that. Later work of Gatys et al. [4] proposed a style transfer algorithm based on deep convolutional neural network VGG [14]. This algorithm was not tied to specific style. Instead the style was specified by a separate style image. Key discovery of Gatys et al. was to find representation for content and style based on activations inside the convolutional neural network. Thus content image could produce target content representation and style image could produce target style representation and for any image we could measure its deviation in style and content, captured by content and style losses. Proposed approach was to find an image giving minimal weighted sum of the content and style loss. Stylization strength was possible by adjusting the weighting factor besides the style loss.

However, algorithm of Gatys et al. required computationally expensive optimization taking several minutes even on modern GPUs. To overcome this issue Ulyanov et al. [16] and Johnson et al. [9] proposed to train a transformer network for fast stylization. The content image there was simply passed through the transformer network for stylization. The network was trained using a weighted sum of content and style loss of Gatys et al. Thus it was tied to specific style and stylization strength, fixed in the loss function, so modification of stylization strength at inference time was not possible.

Further research targeted to propose a transformer network architecture capable of applying different styles simultaneously. Main idea was to hold most of architecture fixed and to vary only particular elements depending on the style. Chen et al. [2] used separate convolution filter for each style. Dumoulin et al. [3] used different parameters of instance normalization, but these coefficients still needed to be optimized. Ghiasi et al. [5] proposed to predict these parameters using a separate style prediction network, thus omitting expensive optimization. However, the problem of stylization strength control was not addressed in these works.

In alternative line of research stylization was performed by passing a content image through an autoencoder with linear scaling of intermediate image representation. Huang et al. [8] applied channel-wise scaling forcing channel-wise means and variances of the content image to match those of the style image. Li et al. [11] extended this approach by matching means and whole covariance matrix of activations at different channels instead. Moreover, in their approach a content image was passed through a sequence of deep and shallow autoencoders and content representation was adjusted in each of them. Stylization strength was possible in these methods by targeting a weighted combination of the content and style moments: higher coefficient besides the style moment imposed more style and vice versa.

Generative adversarial networks [7] (GANs) were also successfully applied to style transfer, for instance, in [1,18]. The difference to the framework considered in this paper is that GANs retrieve style from multiple style images instead of just one.

3 Real-Time Style Transfer with Strength Control

3.1 Baseline Method

Our real-time style transfer with strength control architecture is built upon the baseline method – real-time style transfer of Johnson et al. [9] with minor improvements: batch-normalization layers are replaced with instance normalization layers, following [17], and transposed convolutions replaced by nearest neighbor upsampling and ordinary convolutions to omit checkerboard artifacts, following [12]. Other specifications of layers are not changed, training details are also fully reproduced except that it is empirically found that 80K images are sufficient for convergence of the transformer network. In this method a content image x_c is stylized by passing it through a transformer network $T_w(x_c)$ pretrained to reproduce given style with fixed stylization strength.

3.2 Proposed Extension

Proposed method is built upon the structure of the baseline method. Stylization is performed by passing a content image x_c through the transformer network: $x = T_w(x_c, \alpha)$, where parameter $\alpha \geq 0$ specifies stylization strength. During training transformer network is adapted to produce results for different α, so the user can adjust it at inference in real time. Building blocks of the proposed algorithm remain from the baseline, except residual blocks. In [9] traditional residual block $i = 1, 2, ...5$ for input u produces the output given by the sum of identity transformation and a non-linear transformation: $u + f_i(u)$. Our modified residual block i outputs $u + \gamma_i f_i(u)$, where

$$\gamma_i = 2\frac{|\alpha\beta_i|}{1 + |\alpha\beta_i|}, \tag{1}$$

and β_i is a block-specific trainable parameter, allowing the network to better distribute strength of stylization along blocks. Renormalization (1) is performed to ensure that for any α and β_i resulting factor $\gamma_i \in [0, 2)$ does not dominate the identity connection, allowing the network to propagate gradients well during training. Higher α increases the impact of the non-linear transformation $f_i(u)$ on the final result. Since $f_i(u)$ is responsible for applying style and identity skip-connection leaves input image as it is, γ_i and thus α, naturally control the amount of style added to the input.

The structure of the proposed method is shown on Fig. 1 with the new components, compared to the baseline method, highlighted in red.

3.3 Training

For given stylization strength α our method and the baseline method were trained using conventional style transfer loss from [9], consisting of content, style and total variation components:

$$\mathcal{L}(x, x_c, x_s, \alpha) = \mathcal{L}_c(x, x_c) + \alpha v_s \mathcal{L}_s(x, x_c) + v_{TV}\mathcal{L}_{TV}(x),$$

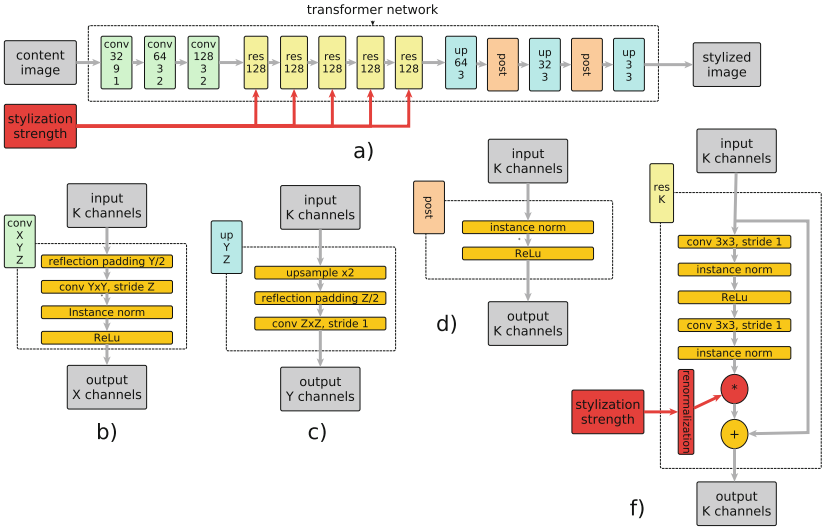

Fig. 1. Architecture of real-time style transfer with stylization strength control, differences from the structure of Johnson are highlighted in red. (a) transfer network (b)–(f) detailed schemes of convolutional, residual, upsampling and post-processing blocks respectively. (Color figure online)

applied to content image x_c, style image x_s and resulting image x with fixed predefined weights v_s and v_{TV}, controlling impact of the corresponding components. Separate baseline model was trained for each α from a grid, and our method used only a single model. To force adjustment of our method to different α, this parameter was sampled uniformly from the grid $[0, 0.1, \ldots 10.0]$ for each mini-batch. Sampling from exponential(1) distribution was also considered, but it didn't provide enough variance to yield sufficiently different results for different α.

Content images were taken from MS COCO 2014 training dataset [19], consisting of 80K images with a batch size 16. Each image was resized and cropped to 256×256. One epoch through the dataset was enough for convergence, and training one model took around 45 min on two NVIDIA GeForce GTX 1080 GPUs. We used Adam optimizer with learning rate 10^{-3}. Total variation strength is set to 10^{-5}.

4 Experiments

4.1 Qualitative Comparison with Real-Time Style Transfer

A complete implementation of our approach in pytorch with pretrained models is available for download[1]. By applying different styles to various content images in

[1] https://github.com/victorkitov/style-transfer-with-strength-control.

our experiments, it is observed that real-time style transfer of Johnson et al. [9], being a very powerful model, overfits to the target loss function. This results in sporadically appearing local artifacts when applying generator model trained with small stylization strength. Our model is based upon framework of Johnson et al. but is less flexible due to necessity to apply style with a whole range of different strengths. This additional constraint serves as regularization and reduces overfitting, alleviating observed artifacts during style transfer. It can be seen on Figs. 2 and 3 where our method gives visually more pleasing results without salient artifacts of the baseline model. These findings are consistent for different content and style images.

4.2 Quantitative Comparison with Real-Time Style Transfer

To compare the results quantitatively we consider 5K content images from MS COCO 2017 validation dataset resized and cropped to 256×256 and calculate average loss by the baseline method of Johnson et al. [9] and by our method for each style out of 8 styles from [10]. Average style loss is approximately 15 times bigger than average content loss and total variation loss is several orders of magnitude less and thus removed from consideration. Next we calculate average ratio between our method loss and baseline method loss as well as standard deviation of this ratio along different styles. Figure 4 shows ratios for total loss, content loss and style loss. It can be seen that our method closely reproduces total loss - it is close to the baseline total loss for style strength greater or equal to 1 and increases only 2 times for style strength dropping to 0.1. Style loss stays very close to the baseline in all cases, so change in the total loss occurs due to significant difference in the content loss. Baseline method indeed is able to achieve lower content loss by preserving content image in all pixels except local regions with strongly expressed style which appears as an artifact and is an undesirable property of the baseline stylizer.

4.3 User Evaluation Study

Proposed method and the baseline method of Johnson et al. [9] were compared on the representative set of 4 content images and 8 style images [10]. Each content was stylized using every style with stylization strength randomly chosen from $[0.1, 0.3, 0.6, 1, 3, 5, 10]$, giving a set of 32 stylizations. 9 respondents were sequentially shown pairs of stylizations in the same setup for our and the baseline method. For each pair they were asked to select stylization they like more. To omit position bias, stylizations of the two methods were placed in random order.

Table 1 gives a summary of the results. These results suggest that the proposed method gives visually more pleasing results than the baseline method of Johnson et al. in 2/3 of cases. This is an expected result since mentioned above artifacts of the baseline method appear consistently in cases of stylization with small strength.

Fig. 2. Johnson et al. [9] vs. our method. Proposed algorithm uses single generator network, while baseline needs separate network for each stylization strength. This single generator is enough to stylize with different strength. The method of Johnson et al. frequently generates local artifacts when performing stylization with small strength (second row). Proposed algorithm alleviates these artifacts and produces more pleasing results.

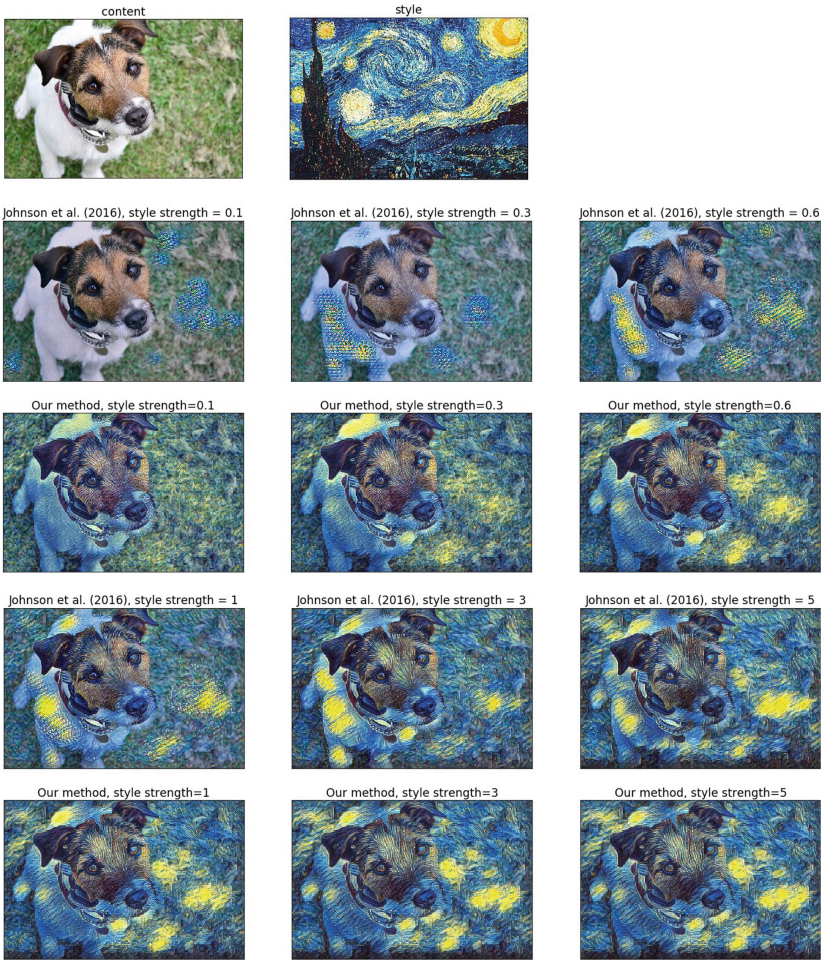

Fig. 3. Johnson et al. [9] vs. our method. Proposed algorithm uses single generator network, while baseline needs separate network for each stylization strength. This single generator is enough to stylize with different strength. The method of Johnson et al. frequently generates local artifacts when performing stylization with small strength (second row). Proposed algorithm alleviates these artifacts and produces more pleasing results.

4.4 Qualitative Comparison with Other Existing Methods

Since AdaIn [8] and universal style transfer [11] enable style strength control during inference, we present qualitative comparisons of our and their stylizations on Figs. 5 and 6. Style strength for these methods is controlled by special parameter which is limited to $[0, 1]$ interval. So we adapt style strength levels of our method to qualitatively match parameter levels of the compared methods.

Table 1. Summary of the user evaluation study.

Total number of image pairs	32
Total number of respondents	9
Total number of responses	288
Number of responses when the proposed method was better than the baseline	192
The same as proportion	66.6%
Number of images that were better rendered by the proposed method	21
The same as proportion	65.6%

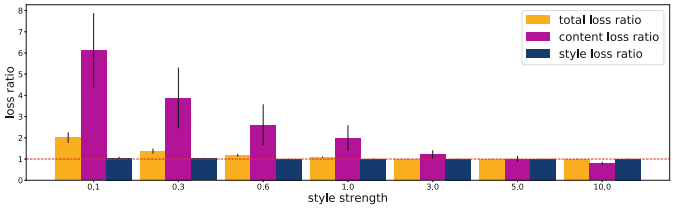

Fig. 4. Ratio of our loss function to the Johnson et al. 2016. Loss function for total, content and style loss evaluated for different style strengths.

Comparisons on Figs. 5 and 6 conform well with the structures of the methods. AdaIn applies style transfer by matching means and standard deviations of the intermediate content image representation to that of the style image. Representation is calculated on single layer of the autoencoder. This simple operation does not allow to reproduce style in detail and generates simplified cartoon-like result instead. Universal style transfer applies style by passing content image through a sequence of autoencoders and adapts mean and whole covariance matrix of the intermediate image representation on each of them. This allows to reproduce more characteristics of style but still without fine details due to limited capabilities of linear scaling. Also since universal style transfer applies style by passing content image through a sequence of deep autoencoders, the result becomes comparatively more blurry.

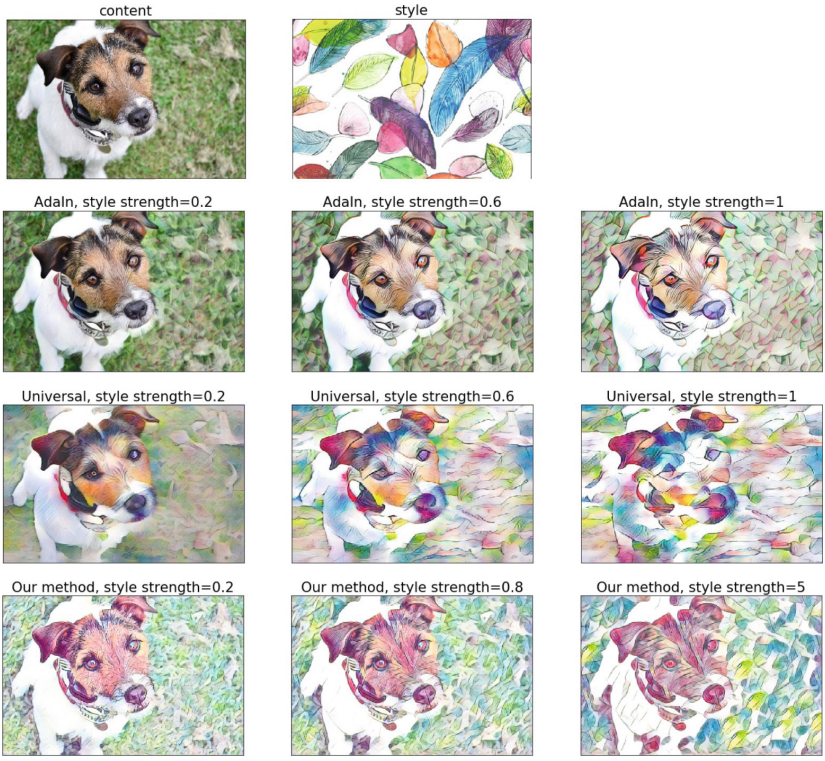

Fig. 5. Qualitative comparison of our method with AdaIn and universal style transfer. Our method better reproduces style details.

Instead of linear scaling our method applies stylization by multiple non-linear transformations which makes it more flexible and allows the method to reproduce style with fine details. Nevertheless, for some applications, such as cartoon-like poster creation, image simplifications obtained by AdaIn and universal style transfer are also desirable properties.

Fig. 6. Qualitative comparison of our method with AdaIn and universal style transfer. Our method better reproduces style details.

5 Conclusion

We have presented an extension to the real-time style transfer of Johnson et al. [9] which allows training a single image transformer network capable of stylization with adjustable stylization strength at inference time. Qualitative and quantitative comparisons show that the proposed architecture is good at applying stylization of different strength and produces results not worse than Johnson et al. Although average content loss obtained by their method is lower, it comes at a price of introducing distracting local artifacts to the stylized image. Proposed algorithm alleviates these artifacts, which may be attributed to the regularization effect of the training procedure forcing the model to solve not a particular task, but a range of tasks. Conducted user study supports our qualitative conclusions that the proposed method gives perceptually more appealing stylization results. Qualitative comparisons with other methods capable of real-time stylization strength control show that our algorithm better preserves details of the style. Thus the proposed algorithm is a viable style transfer solution when real-time control of stylization strength is important.

References

1. Cao, K., Liao, J., Yuan, L.: CariGANs: unpaired photo-to-caricature translation (2018)
2. Chen, D., Yuan, L., Liao, J., Yu, N., Hua, G.: StyleBank: an explicit representation for neural image style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1897–1906 (2017)
3. Dumoulin, V., Shlens, J., Kudlur, M.: A learned representation for artistic style. In: Proceedings of ICLR, vol. 2 (2017)
4. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414–2423 (2016)
5. 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)
6. Gooch, B., Gooch, A.: Non-photorealistic Rendering. AK Peters/CRC Press, New York (2001)
7. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
8. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1501–1510 (2017)
9. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
10. Kitov, V.: Set of content and style images. https://github.com/victorkitov/style-transfer-with-strength-control. Accessed 1 Apr 2019
11. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. In: Advances in Neural Information Processing Systems, pp. 386–396 (2017)

12. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016). https://doi.org/10.23915/distill.00003. http://distill.pub/2016/deconv-checkerboard
13. Rosin, P., Collomosse, J.: Image and Video-Based Artistic Stylisation, vol. 42. Springer, London (2012)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
15. Strothotte, T., Schlechtweg, S.: Non-photorealistic Computer Graphics: Modeling, Rendering, and Animation. Morgan Kaufmann, San Francisco (2002)
16. Ulyanov, D., Lebedev, V., Vedaldi, A., Lempitsky, V.S.: Texture networks: feed-forward synthesis of textures and stylized images. In: ICML, vol. 1, p. 4 (2016)
17. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: the missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
18. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)
19. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48