Depth-Aware Arbitrary Style Transfer Using Instance Normalization

Victor Kitov[1,2][0000−0002−3198−5792], Konstantin Kozlovtsev[1], and Margarita Mishustina[1]

[1] Lomonosov Moscow State University, Moscow, Russia
[2] Plekhanov Russian University of Economics, Moscow, Russia
v.v.kitov@yandex.ru, ko-sova@yandex.ru,
margarita_mishustina_112@mail.ru
https://victorkitov.github.io

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) show that traditional style transfer methods of Gatys *et al.* (2016) and Johnson *et al.* (2016) fail to reproduce the depth of the content image, which is critical for human perception. They suggest to preserve the depth map by additional regularizer in the optimized loss function, forcing preservation of the depth map. However these traditional methods are either computationally inefficient or require training a separate neural network for each style. AdaIN method of Huang *et al.* (2017) 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 by applying variable stylization strength. Qualitative analysis and results of user evaluation study indicate that the proposed method provides better stylizations, compared to the original AdaIN style transfer method.

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 [5,16,14] 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. Ulyanov *et*

al. [17] and Jonson *et al.* [7] proposed a real-time style transfer algorithm by passing a content image through a pretrained fully convolutional transformer network. Their methods required training a separate transformation network for each new style. Work of Liu et. al (2017) [10] highlighted an issue with traditional style transfer methods that they failed to reproduce the depth map of the content image, that was critical for human perception of the result. To address this issue they extended traditional methods [2] and [7] with a regularizer, forcing preservation of the depth map of the content image. This yielded significant improvement of style transfer rendering quality but required computationally complex algorithm, requiring either solving high dimensional optimization problem for each content-style pair or fitting a separate transformer network for each style. Later architectures, such as AdaIN [6] and other ([3], [8]), 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 the work a depth aware AdaIN method extension (DA-AdaIN for short) is proposed that allows to preserve the depth map of the content during stylization, as shown on fig. 1d by applying style with spatially variable strength: more close regions, standing for foreground, are stylized less, and more distant regions, standing for background, are stylized more.

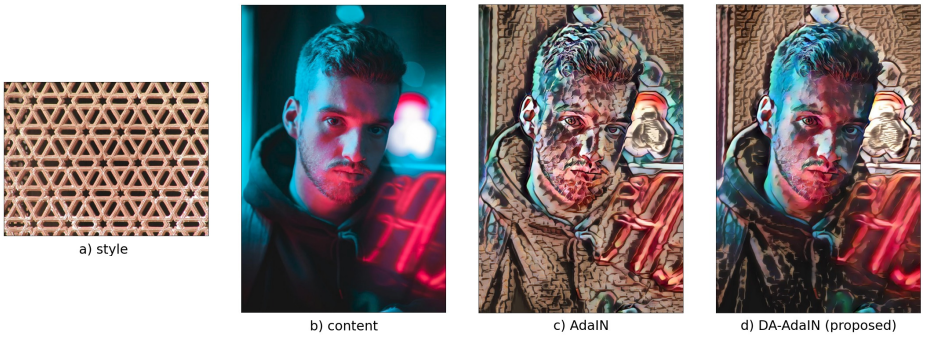

Fig. 1. AdaIN and proposed Depth Aware AdaIN method comparison.

Since style transfer does not yet have conventional objective criteria of quality, we judge which method is better by means of aggregated user preferences. Qualitative analysis suggests that proposed modification leads to rendering quality improvement. User study confirms that proposed algorithm gives on average better results.

The remainder of the paper is organized as follows. In section 2 two recent depth estimation methods are compared and the best one is used in later analysis. Section 3 describes standard AdaIN method and our proposed modification. Section 4 provides qualitative analysis of the proposed method, its dependency on major parameters and results of a user study, where AdaIN and proposed methods are compared. Finally section 5 concludes.

2 Depth estimation

2.1 Methods

Monocular depth estimation is a problem of finding a depth map $D \in \mathbb{R}^{W \times H}$ for arbitrary color image $I \in \mathbb{R}^{W \times H \times 3}$. Since I is a color image, it has three channels, standing for red, green and blue color intensities. D is a single channel image, having the same width W and height H as I, with $D(x, y)$ equal to the distance of pixel $I(x, y)$ to the camera.

For the purposes of style transfer we are interested to discriminate between central objects, that are more close to the camera, from background objects, that are more distant. So absolute accuracy of depth prediction is not as important as relative accuracy.

We compare two recent methods for monocular depth estimation - *monodepth2* [4] and *MiDaS* [13], using official public implementations of both.

MiDaS is a supervised model in contrast to monodepth2, which is self-supervised, which means that it did not use true depth values during training. Monodepth2 was trained on single KITTI 2015 dataset [11], covering outdoor scenes, taken by the camera on the car. MiDaS was trained on five different datasets, covering indoor and outdoor scenes with static and dynamic objects in various contexts.

MiDaS has a single realization, whereas monodepth2 has nine: mono-640x192, stereo-640x192, mono+stereo-640x192, mono-1024x320, stereo-1024x320, mono+stereo-1024x320, mono-no-pt-640x192, stereo-no-pt-640x192, mono+stereo-no-pt-640x192. They differ in training data used (mono, stereo or both), resolution of training data and weights initialization.

2.2 Comparison

Since style transfer may be applied to arbitrary images, we need a depth estimation method that is robust across different types of scenes. Qualitative check for random images shows significant superiority of the MiDaS method, as can be seen on fig. 2.

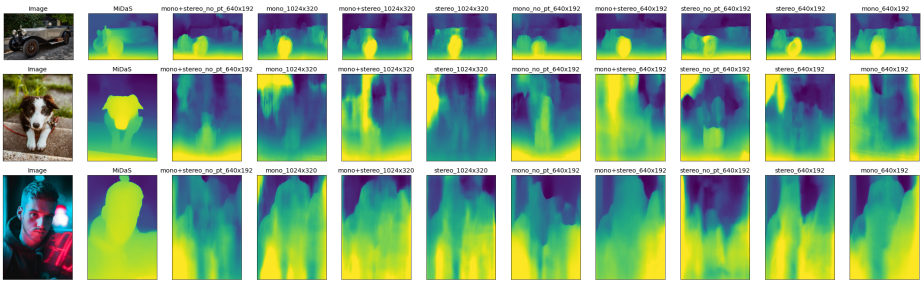

Fig. 2. Qualitative comparison of *MiDaS* [13] (2nd column) and different realizations of *monodepth2* [4] depth estimation methods (columns 3-11). MiDaS is robust to different scenes, whereas monodepth2 has poor generalization for non-road objects.

To compare methods quantitatively we apply them on the test subset of the DIW dataset[1], having diverse kinds of images. Neither of the methods used this dataset for training. The test subset used contains 73983 images with sparse labels: for each image 2 point locations are randomly selected, and an indicator is given whether the first location is more close or distant to the camera, than the second point. We apply depth prediction methods to each image and check, whether relative depth indicator was predicted correctly or not. Accuracy results are reported in table 1. These results confirm that MiDaS is more accurate depth estimation model for images of general kind, so we will use this method in later analysis. This is an expected result since MiDaS was trained in supervised way on a number of diverse datasets.

Table 1. Relative depth prediction accuracy for MiDaS [13] and different realizations of monodepth2 [4] depth prediction methods on the DIW test dataset [1].

Method	Accuracy
MiDaS	**0.87**
mono+stereo-1024x320	0.69
mono+stereo-640x192	0.70
mono+stereo-no-pt-640x192	0.65
mono-1024x320	0.70
mono-640x192	0.71
mono-no-pt-640x192	0.65
stereo-1024x320	0.67
stereo-640x192	0.66
stereo-no-pt-640x192	0.62

3 Style transfer

3.1 AdaIN Method

AdaIN method [6] is a recent powerful style transfer method, allowing to stylize any content image I_c by any style image I_s in real-time without any complex optimizations. Stylization result $\hat{I}$ is obtained by

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

where $f(\cdot)$ is an encoder (taken as first few layers of VGG-19 [15]) and $g(\cdot)$ is a corresponding decoder, trained to match the encoder in producing good stylizations for a representative set of content images (MS COCO [9]) and style images (WikiArt [12])

according to a loss function being a weighted combination of content preservation loss (matching inner VGG-19 representations of content and style) and style preservation loss (matching means and standard deviations of inner representations). For details we refer to original paper [6]. $\mathrm{AdaIN}(x, y)$ is a variant of instance normalization [17], where instance normalization parameters are taken from the style image representation.

Define encoder representations

$$x = f(I_c), \quad x \in \mathbb{R}^{C \times H_c \times W_c}$$
$$y = f(I_s), \quad y \in \mathbb{R}^{C \times H_s \times W_s}$$

Then $\mathrm{AdaIN}(x, y) \in \mathbb{R}^{C \times H_c \times W_c}$ and is defined as

$$\mathrm{AdaIN}(x, y)_{cij} = \sigma_c(y) \left(\frac{x_{cij} - \mu_c(x)}{\sigma_c(x)} \right) + \mu_c(y), \tag{1}$$

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

$$i = 1, 2, ...H_c; \quad j = 1, 2, ...W_c; \quad c = 1, 2, ...C; \tag{3}$$

3.2 Proposed Extension

Standard AdanIN method applies style uniformly across content image. To improve rendering quality of style transfer we propose to apply style with different strength in different regions of content image depending on their proximity to the camera. Closer regions we consider foreground, that needs to be preserved more, so we stylize it less. And vice versa more distant regions we consider background, that can be stylized more.

Uniform stylization strength control can be controlled by hyperparameter $\alpha \in [0, 1]$ in the following formula:

$$\hat{I} = g(\alpha f(I_c) + (1 - \alpha)\, \mathrm{AdaIN}(f(I_c), f(I_s))), \tag{4}$$

since $f(I_c)$ is the original unmodified content encoder representation, whereas $\mathrm{AdaIN}(f(I_c), f(I_s))$ is fully stylized encoder representation.

Since we are interested in spatially variable strength control, we apply modified formula

$$\hat{I} = g(P \odot f(I_c) + (1 - P) \odot \mathrm{AdaIN}(f(I_c), f(I_s))), \tag{5}$$

where $P \in \mathbb{R}^{H_c \times W_c}$ is stylization strength map with strength values for each spatial position of content encoder representation and $\odot$ denotes element-wise multiplication repeated for every channel:

$$\{P \odot F\}_{cij} = P_{ij} F_{cij}$$

Algorithm 1 shows steps for computing stylization strength map P in formula 5. MiDaS algorithm produces proximity map straight away, so for it steps 1,2 are omitted. Max, min and mean operations are produced over all spatial positions and produce a scalar.

Algorithm 1 Styliztation strength map estimation.

Input: content image I_c, monocular depth estimation algorithm, size of content encoder representation $f(\cdot)$ $H_c \times W_c$, offset $\varepsilon \geq 0$, prominence $\beta \geq 0$.

Output: Stylization strength map P.

1: Get depth map D for content image I_c
2: Get proximity map $P = \max D - D$
3: Rescale P to content encoder representation size $H_c \times W_c$
4: $P := (P - \min P)/(\max P - \min P)$
5: $P := P - \text{mean } P$
6: $P := 1/(1 + \exp(-\beta P))$
7: $P := \min\{P, 1 - \varepsilon\}$

Step 4 ensures that proximity has spread in [0,1] interval. Step 6 controls contrast of the depth map by hyperparameter β: higher β corresponds to more prominent changes in the depth map around its mean and $\beta = 0$ makes depth map constant converting proposed algorithm to standard AdaIN. Step 7 constrains proximity map from above by $1 - \varepsilon$. Hyperparameter ε controls the minimal offset from the camera to regions on the image.

For stylization pre-trained AdaIN encoder/decoder [15,6] and pre-trained depth network [13] 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. We name our algorithm *Depth Aware Adaptive Instance Normalization* (*DA-AdaIN* for short).

4 Style Transfer Evaluation

4.1 Dependence on major parameters

Proposed algorithm has two hyperparameters: $\beta > 0$ controls prominence of proximity map around its mean value and $\varepsilon \in [0, 1]$ controls minimal offset of the image regions from the camera. To study impact of these parameters on the stylization result we will use content and style images, shown on fig. 3

Fig. 3. Style transfer result depending on depth contrast parameter β.

Fig. 4 shows how style transfer output depends on contrast parameter β. Higher values increase contrast (spread) of the proximity map values, while ensuring that they fall inside $[0, 1]$ interval.

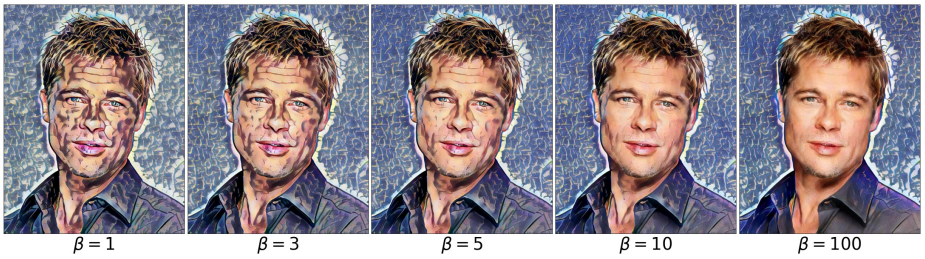

Fig. 4. Style transfer result depending on depth contrast parameter β, $\varepsilon = 0$.

Fig. 5 shows dependency of style transfer results on proximity offset parameter ε. The lower is this offset, the closer proximity values may approach one in certain regions, forcing for that regions more content reconstruction and less style transfer. Higher values of ε ensure that all image regions maintain certain distance from the camera and higher minimal impact of style transfer is ensured.

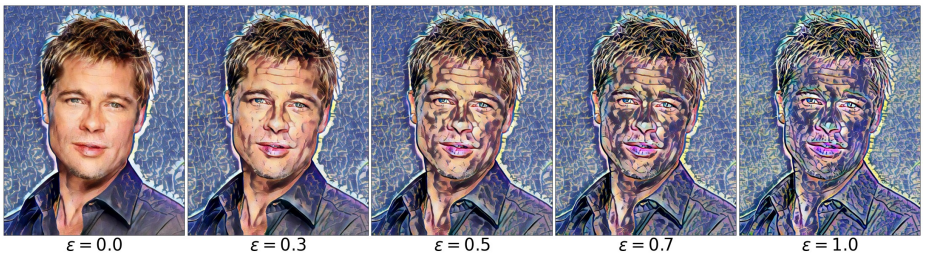

Fig. 5. Style transfer result depending on proximity offset parameter ε, $\beta = 20$.

4.2 Qualitative Comparison With AdaIN method

Side-by-side comparisons of style transfer results by standard AdaIN method and proposed modification DA-AdaIN is visualized on fig. 6. For DA-AdaIN we used $\varepsilon = 0.15$ and $\beta = 20$. Comparisons show that proposed method is capable to detect more close objects and highlight them by applying style transfer with smaller strength. More close objects generally are more important for the viewer and this strategy allows to preserve

them better by less expressed stylization, which brings rendering improvement. However if this approach is used too strongly, it may create a noticeable disagreement between foreground and background rendering, as may be seen on the last row of fig. 6 where a huge proximity contrast between the foreground (the dog) and the background (the grass) forced the foreground look too photorealistic in strongly stylized context. To alleviate this issue we suggest to increase offset ε or decrease contrast β.

4.3 User Evaluation Study

Procedure. To provide a more general comparison of style transfer methods we conduct a user evaluation study, where 18 users were asked to pass a survey. The survey consisted of 20 image pairs, corresponding to stylizations by AdaIN and DA-AdaIN methods presented in random order, and the users had to select for each pair a stylization which they liked more. 360 responses were collected. For a set of different style and content images all possible stylizations were generated. Contents were selected to contain objects of different proximity to the camera, otherwise results between the two methods were indistinguishable. A random subset of 20 results was selected for the survey. Contents were resized so that their smaller side is 1000 pixels and styles were resized to ensure that their smaller side is 300 pixels. We did not tell the respondents anything about the depth preservation concept and our algorithm details. Although performance of our method could be further improved by finetuning parameters ε and β for each individual image, we fixed them to general reasonable values $\varepsilon = 0.15$ and $\beta = 20$, to put proposed method in equal position with the baseline.

Results. The results of user study evaluation study are presented on table 2. Our method is preferred moderately more often than existing AdaIN method and this difference is statistically significant with 99% confidence for exact binomial test.

Table 2. Results of user evaluation study

Experiment	Ours vs AdaIN
# image pairs	20
# respondents	18
# responces	360
# votes for proposed method	207
proportion of votes for proposed method	**57.5%**
std. deviation of proportion	2.6%
p-value (exact binomial test)	0.0026

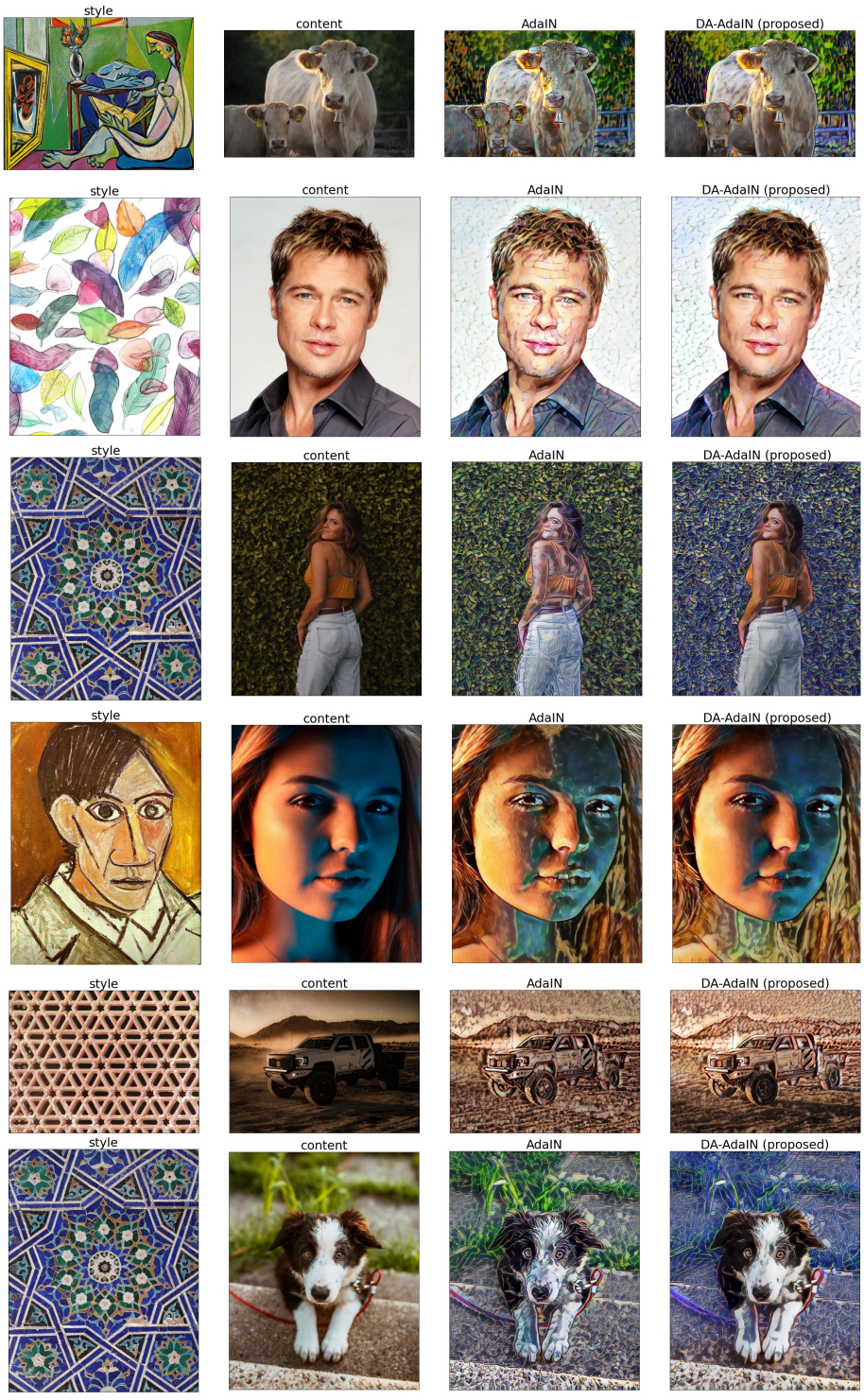

Fig. 6. Comparison of style transfer results for AdaIN and proposed DA-AdaIN methods

.

Discussion. During the study it was found that DA-AdaIN was not very sensitive to proximity variability, so the contrast in variability had to be additionally increased by step 6 of algorithm 1. For contents with very close objects to the camera, proximity became close to one for those objects and they received almost no stylization, so offset on step 7 of algorithm 1 was introduced to maintain certain guaranteed level of stylization for all parts of the image. These modifications ensured better rendering quality on average. $\beta = 20$ and $\varepsilon = 0.15$ are recommended. For particular content and style pair result may be improved even more by manual tuning of β and ε parameters.

5 Conclusion

An extension to AdaIN method, allowing to preserve depth information from the content image, is proposed. 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 the objects on the stylized image and results of the user evaluation study confirm that depth preservation is important for users, making them prefer our method more often than conventional AdaIN method.

References

1. Chen, W., Fu, Z., Yang, D., Deng, J.: Single-image depth perception in the wild. In: Advances in neural information processing systems. pp. 730–738 (2016)
2. 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)
3. 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)
4. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth prediction (October 2019)
5. Gooch, B., Gooch, A.: Non-photorealistic rendering. AK Peters/CRC Press (2001)
6. 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)
7. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016)
8. 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)
9. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
10. Liu, X.C., Cheng, M.M., Lai, Y.K., Rosin, P.L.: Depth-aware neural style transfer. In: Proceedings of the Symposium on Non-Photorealistic Animation and Rendering. p. 4. ACM (2017)
11. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3061–3070 (2015)
12. Nichol, K.: Painter by numbers, wikiart (2016)

13. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. arXiv:1907.01341 (2019)
14. Rosin, P., Collomosse, J.: Image and video-based artistic stylisation, vol. 42. Springer Science & Business Media (2012)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. Strothotte, T., Schlechtweg, S.: Non-photorealistic computer graphics: modeling, rendering, and animation. Morgan Kaufmann (2002)
17. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)