Fast photographic style transfer based on convolutional neural networks

Li Wang, Nan Xiang, Xiaosong Yang*, Jianjun Zhang
Bournemouth University

ABSTRACT

The techniques for photographic style transfer have been researched for a long time, which explores effective ways to transfer the style features of a reference photo onto another content photograph. Recent works based on convolutional neural networks present an effective solution for style transfer, especially for paintings. The artistic style transformation results are visually appealing, however, the photorealism is lost because of content-mismatching and distortions even when both input images are photographic. To tackle this challenge, this paper introduces a similarity loss function and a refinement method into the style transfer network. The similarity loss function can solve the content-mismatching problem, however, the distortion and noise artefacts may still exist in the stylized results due to the content-style trade-off. Hence, we add a post-processing refinement step to reduce the artefacts. The robustness and effectiveness of our approach has been evaluated through extensive experiments which show that our method can obtain finer content details and less artefacts than state-of-the-art methods, and transfer style faithfully. In addition, our approach is capable of processing photographic style transfer in almost real-time, which makes it a potential solution for video style transfer.

CCS CONCEPTS

• **Computing methodologies** → **Computational photography**; **Image processing**; *Image representations*;

KEYWORDS

image processing, photographic style transfer, deep learning, real-time

ACM Reference Format:
Li Wang, Nan Xiang, Xiaosong Yang*, Jianjun Zhang. 2018. Fast photographic style transfer based on convolutional neural networks. In *CGI 2018: Computer Graphics International 2018, June 11–14, 2018, Bintan Island, Indonesia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3208159.3208165

* corresponding author.

CGI 2018, June 11–14, 2018, Bintan Island, Indonesia

ACM ISBN 978-1-4503-6401-0/18/06…$15.00
https://doi.org/10.1145/3208159.3208165

1 INTRODUCTION

Image stylization is a research topic with great impact and strong appeal in social media, where people treat photo sharing and exchanging as important social activity in daily life. Many methods for image stylization have been proposed in the last decade, including colour transfer, texture transfer and style transfer. Recently, a series of artistic style transfer methods [3, 4, 9, 10, 12, 13, 16, 21–23, 31, 35, 36, 38, 41] based on convolutional neural networks (CNN) reveal a new solution for image style transformation. Gatys *et al.* [11] firstly proposed a Neural-Style method to transfer artistic features from artistic painting to regular content images. It achieves visually pleasing results by measuring the differences between the reference style picture and the content image. Filter responses extracted from network layers are regarded as feature representations of the inputs [27]. The loss functions in these layers minimize the representation distances leading to a new image combining the rough spatial structures from the content image and the reference artistic features (including texture and colour). Inspired by the results, Johnson *et al.* [17] proposed a Fast-Neural-Transfer method which achieves visually pleasing results with better reconstructed fine details and edges than the Neural-Style [11] method. In addition, the efficiency benefit of embedding a feed-forward transformation network with perceptual loss functions inspires plenty of researches such as Texture-nets [35], MSG-net [41] and Multi-texture-synthesis [23]. Generative adversarial networks with Markov random fields to synthesize texture from art reference pictures and content details from another photograph also have recently attracted a lot of attention, such as the Markovian Generative Adversarial Networks (MGA-networks) proposed by Li *et al.* [21] achieving satisfactory results.

All these CNN-based methods pursue to transfer style from a reference painting to a content photograph. However, it often fails to transfer the photorealistic style when the reference image is photographic because of content-mismatching and distortions. For photographic style transfer, Luan *et al.* [26] proposed a Deep-Photo-Style-Transfer (DPST) method to solve the problem. It adds a photorealism regularization term based on locally affine colour transformations to prevent distortions, and uses semantic segmentation to avoid the content-mismatching problem. The content spatial structures are preserved in many situations, but details especially the exact edges are erased when semantic segmentation is inaccurate or contains overlapping areas. Moreover, extra computation of Matting Lapacian matrix and semantic segmentation consumes a lot of computing time even for low resolution images.

In this paper, we present a new technique to convert prior artistic style transfer methods into photographic style transfer, which improves the photorealism attribute of the stylized results. Without semantic segmentation, our method introduces a similarity loss function to solve the content-mismatching problem, and a

Figure 1: The photorealism of stylized result is lost. Note that the reconstructed content contains the unexpected distorted details such as shapes of buildings in (a), and the stylized result also suffers from the content-mismatching problem such as the undesired blended region in the top right area of (b). These problems make the stylized result look unrealistic. Examples are from Luan *et al.* [26]

post-processing technique to reduce potential distortion and noise artefacts. The similarity loss function reconstructs finer details of content photographs and constrains the content match between reference style and content images. The post-processing refinement technique extracts the colour without the details from stylized result, and combines it with the details of content input. Distortion and noise artefacts will be eliminated after the refinement step. Intergrating the mentioned above mehtods into prior artistic style transformation networks, our method achieves almost real-time performance. This advantage makes our approach a good option for real-time application such as video style transfer.

There are two major contributions in this paper:

1. We introduce a similarity loss function and post-processing refinement technique into the CNN-based style transfer frameworks, which transfer style faithfully and produce photorealistic results.

2. We propose a photographic style transfer method with real-time performance, which makes it a potential solution for socical media apps and video style transfer.

2 RELATED WORK

In this section, we will review several representative techniques referring to colour transfer and photographic style transfer. As this paper focuses on only photographic images, the researches about artistic style transfer are not included.

Global Colour Transfer. Colour transfer methods [30] [29] [14] [39] tend to be a global transformation between images. A spatial-invariant transfer technique is applied to handle simple situations, such as global colour move (*e.g.*, sepia) and tone curves (*e.g.*, low or high contrast). For example, Reinhard *et al.* [30] propose a colour shift technique to match the mean and standard deviation between the reference style and content images. It extracts the features from reference style images in a decorrelated colour space. Pieté *et al.* [29] propose a global colour transformation by matching a full 3D colour histogram with a series of 1D histogram. HaCohen *et al.* [14] present a transformation technique based on the global non-linear colour mapping, which relies on the local correspondences between images. Their results are compelling but their approach highly depends on the pairs of photos. For example, the input photos should depict similar scenes with different colours, views and illumination.

Local Style Transfer. Local colour transfer methods are capable of being more expressive and handling a various class of applications such as weather and season change [7, 19], and time-of-day hallucination [7, 32]. Such methods based on spatial colour mapping highly require sparse correspondence guidance from either user input [1, 37] or image segmentation [2, 34, 40]. Results of these algorithms are not precise enough because some pixels can be transferred into inaccurate colours. Recently, some remarkable photorealistic style transfer methods [15, 24, 26, 28] based on CNN provide an additional perspective for style transformation and colour transfer. Luan *et al.* [26] propose a Deep-Photo-Style-Transfer method for photorealistic style transfer, which expanses the Neural-Style algorithm [11]. The solution to content-mismatching and distortions problems depends on the semantic segmentation and precomputed matting Lapacian Matrix [20] of content photo respectively. Mechrez *et al.* [28] use screened Poisson equation to replace the Luan *et al.*'s post-processing step, which can improve the photorealism of Luan *et al.*'s results. More recently, a content-matching strategy between deep features is used on photographic style transfer. Liao *et al.* [24] apply a coarse-to-fine strategy to compute the nearest-neighbour field for accomplishing the spatially-variant and globally coherent colour transfer.

Our method follows directly from the line of work Fast-Neural-Transfer [17]. The Fast-Neural-Transfer algorithm proposes an image transformation feed-forward network to solve the slow optimization problem, where a loss network using stochastic gradient descent is applied to train the feed-forward network. It presents similar qualitative results compared to Gatys *et al.* [11]. However, photorealism loss is still a stranding problem for photographic style transfer because of the distortions and content-mismatching problems. To address these problems, we introduce a similarity loss function into the loss network, and a post-processing refinement technique. The similarity loss function makes sure the content details are well reconstructed and correctly mapped from reference style image to content input. And the post-processing technique prevents potential distortion and noise artefacts.

3 METHOD

The basic architecture of our system consists of two components: an image stylizing feed-forward network $F_W(\cdot)$ and a loss network $L(\cdot)$. Johnson *et al.* use a deep residual CNN as the image stylizing network, which is parameterized by weights W. The Loss Network is the pretrained VGG-16 network [33]. The input images $\vec{x}$ are passed through the image stylizing network, and they are transformed into one output image $\tilde{y}$ via the mapping function $\tilde{y} = F_W(\vec{x})$. For each $\vec{x}$, we have a content target y_c and style target

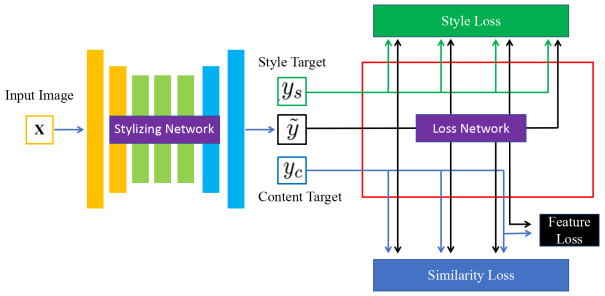

Figure 2: System overview. The system consists of two components: a Stylizing Network and a Loss Network. Orange, green, black and blue rectangles represent an input image, a style target, an output image and a content target, respectively. The style loss, feature loss and similarity loss are defined on the Loss Network. These losses are used to train the Stylizing Network.

y_s. For the loss network, the content target y_c is $\vec{x}$. The training of the image stylizing network pursues weights W which minimizes a weighted total loss function:

$$W = \arg\min_W E_{\vec{x},\{y_c,y_s\}}(\lambda L(\tilde{y}, y_c, y_s)) \tag{1}$$

3.1 Loss Functions for the Loss Network

To clarify the background and our improvement, we split loss functions in our loss network into two cagetories: VGG loss and our similarity loss based on L1-norm.

We summarize the Fast-Neural-Style algorithm by minimizing the objective function:

$$L_{VGG} = \alpha L_{fea}(\tilde{y}, y_c) + \gamma L_{style}(\tilde{y}, y_s) \tag{2}$$

with:

$$L_{fea}(\tilde{y}, y_c) = \sum_{j=1}^{J} \frac{1}{N_j \times M_j} \|F_j(\tilde{y}) - F_j(y_c)\|_2^2 \tag{3}$$

$$L_{style}(\tilde{y}, y_s) = \sum_{j=1}^{J} \frac{1}{N_j^2} \|G_j(\tilde{y}) - G_j(y_s)\|_F^2 \tag{4}$$

where J denotes the total number of activation layers and j is the j-th layer of our loss network. In each layer, the feature maps have N channels and M size where M is width times height. $F_j[\cdot] \in \mathbb{R}^{N_j \times M_j}$ denotes the feature matrix at j-th layer. $G_j[\cdot] = F_j[\cdot]F_j[\cdot]^T \in \mathbb{R}^{N_j \times N_j}$ denotes the Gramian Matrix, which is the inner product between the vectorized feature maps. α and γ denote the weights to feature loss L_{fea} and style loss L_{style}, respectively. y_c and y_s represnt the content target and style target separately.

Zhao *et al.* [42] present a study that L1-norm loss outperforms L2-norm loss on image reconstruction for the same CNN architecture. Inspired by this, we attempt to use L1-norm loss in several layers to reconstruct precise content spatial structures of the target image. Despite of the L1-norm loss term employed outside of network in [42], we add a similarity preservation loss L_{sim} based on mean absolute error (L1-norm) into the Loss Network. Let *MAE* denote the mean absolute error of the feature maps of $\tilde{y}$ and y_c at jth activation layer of the Loss Network, then the similarity preservation loss is denoted as

Figure 3: The similarity loss function for preventing content-mismatching problem. (a) and (b) are the stylized results through the Loss Network without and with similarity loss respectively. Note that (b) indicates the similarity loss effectively prevents the content-mismatching problem in the stylized result.

$$L_{sim}(\tilde{y}, y_c) = \sum_{j=1}^{J} MAE(F_j(\tilde{y}), F_j(y_c)) \tag{5}$$

where J denotes the number of activation layers for similarity preservation loss in the Loss Network. Figure 3 demonstrate the effect of L_{sim}.

Overall, the total loss of the Loss Network is given by:

$$\begin{aligned} L(\tilde{y}, y_c, y_s) = \alpha L_{fea}(\tilde{y}, y_c) + \beta L_{sim}(\tilde{y}, y_c) \\ +\gamma L_{style}(\tilde{y}, y_s) \end{aligned} \tag{6}$$

where β denotes the weight of similarity loss L_{sim}.

3.2 Post-processing Step

We use L_{sim} to avoid the content-mismatching problem, however, the output result may still show distortion and noise artefacts (c.f. Figure 5a). This is because of the content-style trade-off in Figure 7b. To further reduce the artefacts, we introduce a refinement technique into our approach. We use a 2D edge preserving filter (Recursion Filter) [8] to refine the output image $\tilde{y}$ with guidance of content image x. The refined result O_x is defined as

$$O_x = (x - RF(x, \sigma_s, \sigma_r, x)) + RF(\tilde{y}, \sigma_s, \sigma_r, x) \tag{7}$$

where σ_s denotes the spatial standard deviation and σ_r denotes the range standard deviation for the Recursion Filter. In Figure 5c, the refined result O_x generated by the post-processing step finally reduces the distortion and noise artefacts, and exhibits fine content details.

4 IMPLEMENTATION DETAILS

Our approach ([17]+ours) is based on the feed-forward netowrk, which means loss functions are only applied in the training stage, and post-processing refinement step is only applied in the test stage. For the training process, we train the stylizing network on the MS-COCO dataset [25]. The 80k training images are all resized to 256×256, and the style image is resized to $width = 384$ for 40k iterations using a batch size of 4. The training data is given two epochs. We use Adam [18] with learning rate 1×10^{-3}, and a total variation

Figure 4: The post-processing step can not prevent the content-mismatching problem. In the middle (b), we show 2 insights of (a) and (c) (in that order). Zoom in to compare results. Note that the stylized result (c) preserves well the spatial structures of building (green rectangle), but it can not prevent the unexpected yellow colour regions (red rectangle) caused by content-mismatching. We recommend readers to view the electronic version of pictures.

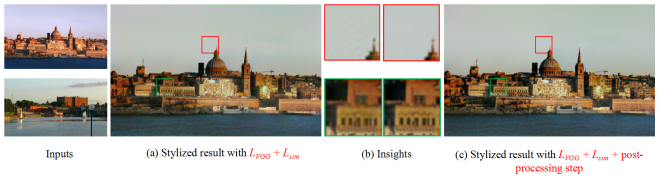

Figure 5: The post-processing step reduces the potential distortion and noise artefacts. In the middle (b), we show 2 insights of (a) and (c) (in that order). Zoom in to compare results. Note that the stylized result (c) prevents the distortion (green rectangle) and noise artefacts (red rectangle), thus exhibits finer details than (a). We recommend readers to view the electronic version of pictures. Examples are from Shih *et al.* [32]

regularization with the strength weight 1×10^{-6}. No weight decay or dropout is used because of the model does not overfit within two epochs. For all the image stylization experiments, we add the similarity layers into relu1_2,relu2_2 and relu3_3 activation layers of the Loss Network. The feature layers and style layers use the default settings of the Fast-Neural-Transfer [17], which are relu3_3 and relu1_2,relu2_2,relu3_3,relu4_3 activation layers of the Loss Network respectively. The hyperparameters of the Loss Network are set as $\alpha = 1.0$ and $\beta = 10.0$ for content reconstruction, and $\gamma = 5.0$ for style transformation. The training takes roughly 2 hours on a single NVIDIA GTX 1080 Ti GPU in the implementation of Torch [6] and cuDNN [5]. For the post-processing refinement step, we use $\sigma_s = 60$ (default in its open source code) and $\sigma_r = 1$ for the Recursion Filter [8]. The effect of σ_r is illustrated in Figure 6.

5 RESULTS

5.1 The Content-style Trade-off

As shown in Figure 7, different values of parameter β directly affect the content-style trade-off. A small β (*e.g.,* (a)) value reconstructs content details worse than other β values. Conversely, a too large β value suppresses the style transfer. For example, the bigger β value tends to remain the colour of house in (c) and (d) as content image does, which actually should be in white colour just like the style image. Hence, we found the best parameter $\beta = 10$ to produce our result and all the other results in this paper.

5.2 Comparison to Previous Work

We introduce the similarity loss function and post-processing refinement step into the baseline artistic style transfer method [17], and transfer the colour of the style image while improving the photorealism of stylized results.

Comparison with baseline artistic style transfer method. In Figure 8, we compare our method to prior representative artistic style transfer network Johnson *et al.* [17] The stylized results obtained from [17] method still suffers from the content-mismatching problem, for example, the sky in the first three rows. Our method also reconstructs finer content details than the baseline(*e.g.*, the fourth and fifth row).

Comparison with global colour transfer methods. Reinhard *et al.* [30] and Pitié. *et al.* [29] are based on the global colour statistics of inputs, which limits their ability to transfer colour between more sophisticated images. For example, in the second row of Figure 9, Reinhard *et al.* and Pitié *et al.* fail to render the sky in black to match the colour of sky in the style image. On the contrary, our method is local and capable of handling more semantic colour transfer.

HaCohen *et al.* [14] propose a NRDC method which relies on a small number of matchable points to estimate the global colour transfer between inputs. Due to this, their method obtains better results than Pitié *et al.*'s (*e.g.*, branches of trees in the third row in Figure 9). However, their method fails to conduct colour transfer between two different scenes (*e.g.*, top two rows in Figure 9). Our method matches colour statistic in different levels of deep feature maps (matching Gramian Matrix at several layers of Loss Network), therefore our results are more accurate than HaCohen *et al.*'s method. For example, in the fourth row of Figure 9, the sky in our result preserves the style of reference image better than HaCohen *et al.*'s, which is too bright in HaCohen *et al.*'s result. Besides, our region of grassland covered in green is more accurate than HaCohen *et al.*'s result.

Comparison of the integration of our method with local photographic style transfer frameworks. Luan *et al.* [26] propose a two-stage photo style transfer method. Its first stage integrates Neual-Style algorithm [11] with semantic segmentation to achieve local object-object colour transfer. The second stage attempts to improve the photorealism of stylized results via a post-processing step, which is based on the Matting Lapacian of [20]. Compared to semantic segmentation, our similarity loss function can not achieve such sophisticated object-to-object style transfer as Luan *et al.*'s method does. However, our post-processing refinement technique may further improve the photorealism of stylized result obtained by Luan *et al.*'s first stage. We use the Recursion Filter [8] rather than [20] to refine the stylized results obtained by Neural-Style with semantic segmentation. In Figure 10, we may notice that our results obtain finer details than Luan *et al.*'s results while preserving the style transfer performance. For example, our refined result preserves finer details of the buildings in the first row and bubbles inside the glass in the fourth row. Moreover, our refined results maintains clearly even the characters on bottle bottom in the third row and better boundaries of cupboards in the

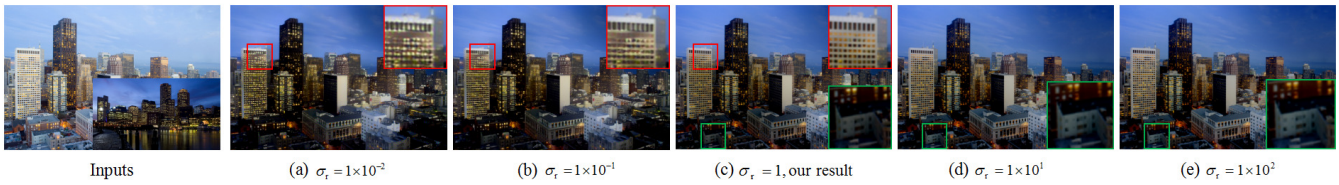

Figure 6: Effect of parameter σ_r for Recursion Filter [8] in the post-processing step. The inputs contains content image and style image (bottom right). Note that a small σ_r value does not reduce noise artefacts (red rectangles) in (a) and (b). In constrast, a too large σ_r value does not keep buildings in dark colour (green rectangles) of (d) and (e), while (c) does. Hence, we use $\sigma_r = 1$ to produce our result and all the other results in this paper. We recommend readers to view the electronic version.

Figure 7: The effect of similarity weight β for content-style trade-off. The transformation result $\tilde{y}$ (b) using parameter $\beta = 10$ preserves finer context of content than smaller β value, for example, the left trees (red rectangle) in (b) are reconstructed with finer details than (a). Moreover, (b) remains the white colour gradient style of house (green rectangle) better than (c) and (d). We conduct a series of experiments with the parameter $\beta = 10$, and obtain almost the same content-style trade-off effect on other images. Hence, we use similarity weight $\beta = 10$ to produce our stylized result $\tilde{y}$ and all the other stylized results in this paper. We recommend readers to view the electronic version of pictures.

bottom row. We recommend readers to view the pictures at full size on a screen. Compared to Liao *et al.* [24], our refined results achieve more faithful style transfer results. For instance, our refined results remain the dark colour gradient of style image in the first and fourth row.

In Figure 11, we compare our method ([17] + ours) to [26] and [24]. Our method may not achieve better style transformation performance than them, but our method is three orders of magnitude faster than theirs while obtaining similar visual transfer appearance.

In Figure 12, we show some comparisons between our proposed approaches ([11]+ours, [17]+ours, [26]+[8]). We may notice that [17]+ours method achieves similar visual appearance compared to [11]+ours, while [26]+[8] obtains more faithful style transfer results than others(*e.g.*, the third and sixth row). Due to the space limit, we show the figure in one colomn.

In Figure 13, we show some examples of failure. Note that the content-mismatching problem may still occur when inputs have very poor content semantic similarity. For example, the kitchen and the nightscape images in the left have big content differences. Additionally, our refinement step may produce stylized results lack of the intensity of colour (*e.g.* apple in Figure 13). This can be fixed by fine-tuning of parameters σ_s and σ_r in the post-processing refinement step.

5.3 Speed Performance

In Table 1, we compare the runtime of baseline methods and ours ([17]+ours) for 256×256 and 512×512 image resolution. The baseline methods include Luan *et al.* [26] and Liao *et al.* [24]. We use the Recursion Filter proposed in [8] as our post-processing refinement step, and the code provided by the authors is implemented in MATLAB with CPU E5 (3.50GHz). All the runtimes exclude the I/O operation (*e.g.* write the file into the disk). For 256×256 resolution, our method achieves a speed up of approximately 4717 and 3118 compared to Luan *et al.* [26] and Liao *et al.* [24], respectively. For 512×512 resolution, our method achieves a speed up of 5808x and 7624x, compared to them, respectively. Our method ([17]+ours) processes 512×512 image at approximately 16 FPS, which makes it feasible to run in real-time or on video.

5.4 User Study

A successful photographic stylized image should look natural to a human observer. Therefore, we conduct a user survey to verify our methods and other four methods. The user survey assesses the photorealism of results and the style faithfulness. There are six methods in total considered in the survey: Reinhard *et al.* [30], Pitié *et al.* [29], Luan *et al.* [26], Liao *et al.* [24] and our methods ([17]+ours, [26]+[8]). Each result image has been shown to human participants who were asked to score the image from 1 to 4. There were only two simple questions: "Does the picture look photorealistic? " and "Do you think the colour looks like the reference style image". For the first question on photorealism, the score on

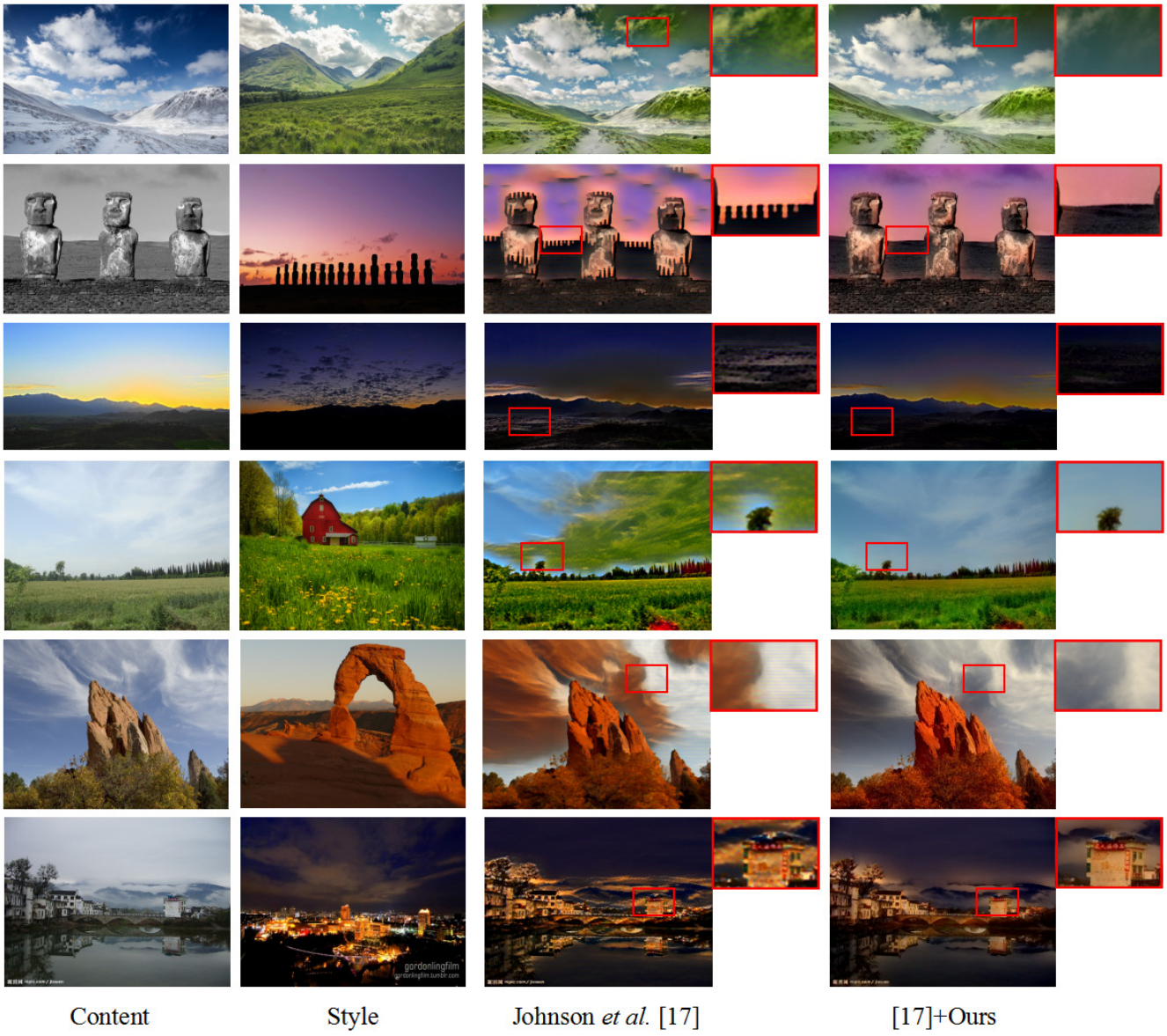

Figure 8: Comparison between baseline artistic style transfer method [17] and ours. All examples are from Luan *et al.* [26]

Table 1: Speed (in seconds) for the baselines and our method.

Image size	Baselines		Our method	Speedup	
	Luan [26]	Liao [24]	[17]+ours	Luan [26]	Liao [24]
256×256	108.510	72.358	0.023	4717x	3118x
512×512	342.723	449.833	0.059	5808x	7624x

a 1-to-4 scale ranging from 'definitely not photorealistic' to 'definitely photorealistic', and only the stylized results were presented to people. For the second question on style faithfulness, the score on a 1-to-4 scale ranging from 'definitely not ' to 'definitely yes' and only corresponding pairs of the stylized results and style images were presented to people. We used 40 images from the dataset of [26] excluding unrealistic inputs. We showed the stylized results to 26 human observers in the survey. We use manually semantic segmentation masks provided by Luan *et al.* for all the results of [26] in this paper.

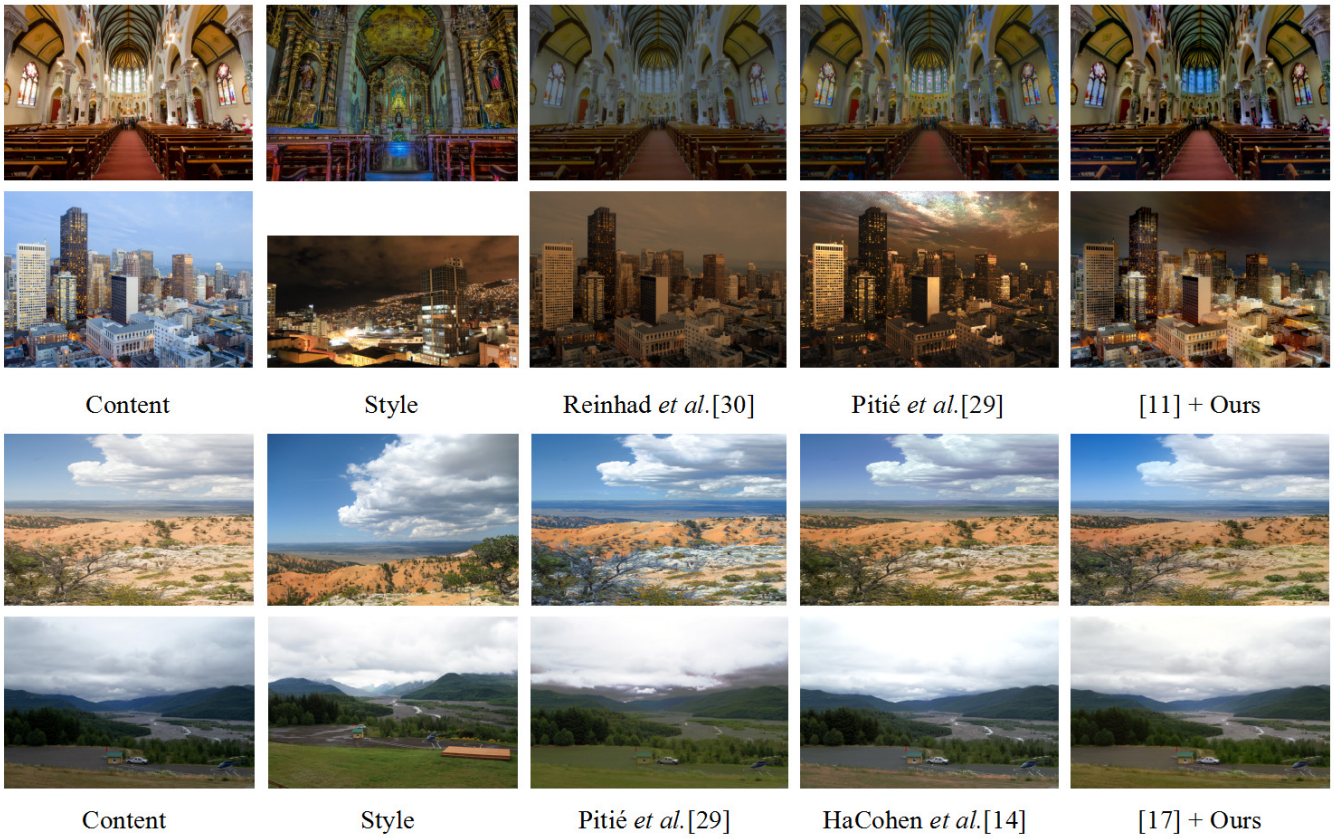

Figure 9: Comparison between global colour transfer methods [30], [29], [14] and ours. Top two examples are from Luan *et al.* [26], and bottom two examples are from HaCohen *et al.* [14].

The average score and standard deviation of each method is shown in Figure 14. For the photorealism, our method ([17]+ours) and Liao *et al.* [24] rank 1 and 2 respectively regarding to photorealism. Pitié *et al.*'s method [29] and Luan *et al.* [26] perform the worst in photorealism, due to some artefacts. For the style faithfulness, Luan *et al.* [26] achieves the highest score among the six methods, because we use manually semantic segmentation masks for [26]. Our refinement step ([8]) slightly declines the style faithfulness of Luan *et al.* [26] but still obtains a higher faithfulness score than Liao *et al.* [24]. Moreover, it significantly improves the photorealism of Luan *et al.*'s [26] results by avoiding the posterization artefacts. Reinhard *et al.* [30] and Pitié *et al.* [29] are the worst in style faithfulness as they are limited to transfer colour for sophisticated images.

6 CONCLUSION

To improve the photorealism of style transformation results, we introduce a similarity loss function and post-processing refinement step into the existing CNN-based artistic style transfer networks. The similarity loss function effectively avoids the content-mismatching problem while reconstructing finer content details, and the refinement step reduces the potential distortion and noise artefacts. Our method can convert prior CNN-based artistic style transformation networks directly into photographic style transfer. The extensive experiments show that our method can obtain finer content details and less artefacts than state-of-the-art methods, and transfer style faithfully. In addition, our approach is capable of processing photographic style transfer in almost real-time, which makes it a potential solution for video style transfer.

ACKNOWLEDGMENTS

The authors would like to thank for the public MATLAB code of the *Recursion Filter* method.

REFERENCES

[1] Xiaobo An and Fabio Pellacini. 2010. User-Controllable Color Transfer. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 263–271.
[2] Benoit Arbelot, Romain Vergne, Thomas Hurtut, and Joëlle Thollot. 2017. Local texture-based color transfer and colorization. *Computers & Graphics* 62 (2017), 15–27.
[3] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. 2017. Stylebank: An explicit representation for neural image style transfer. *arXiv preprint arXiv:1703.09210* (2017).
[4] Tian Qi Chen and Mark Schmidt. 2016. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337* (2016).
[5] Sharan Chetlur, Cliff Woolley, Philippe Vandermersch, Jonathan Cohen, John Tran, Bryan Catanzaro, and Evan Shelhamer. 2014. cudnn: Efficient primitives for deep learning. *arXiv preprint arXiv:1410.0759* (2014).

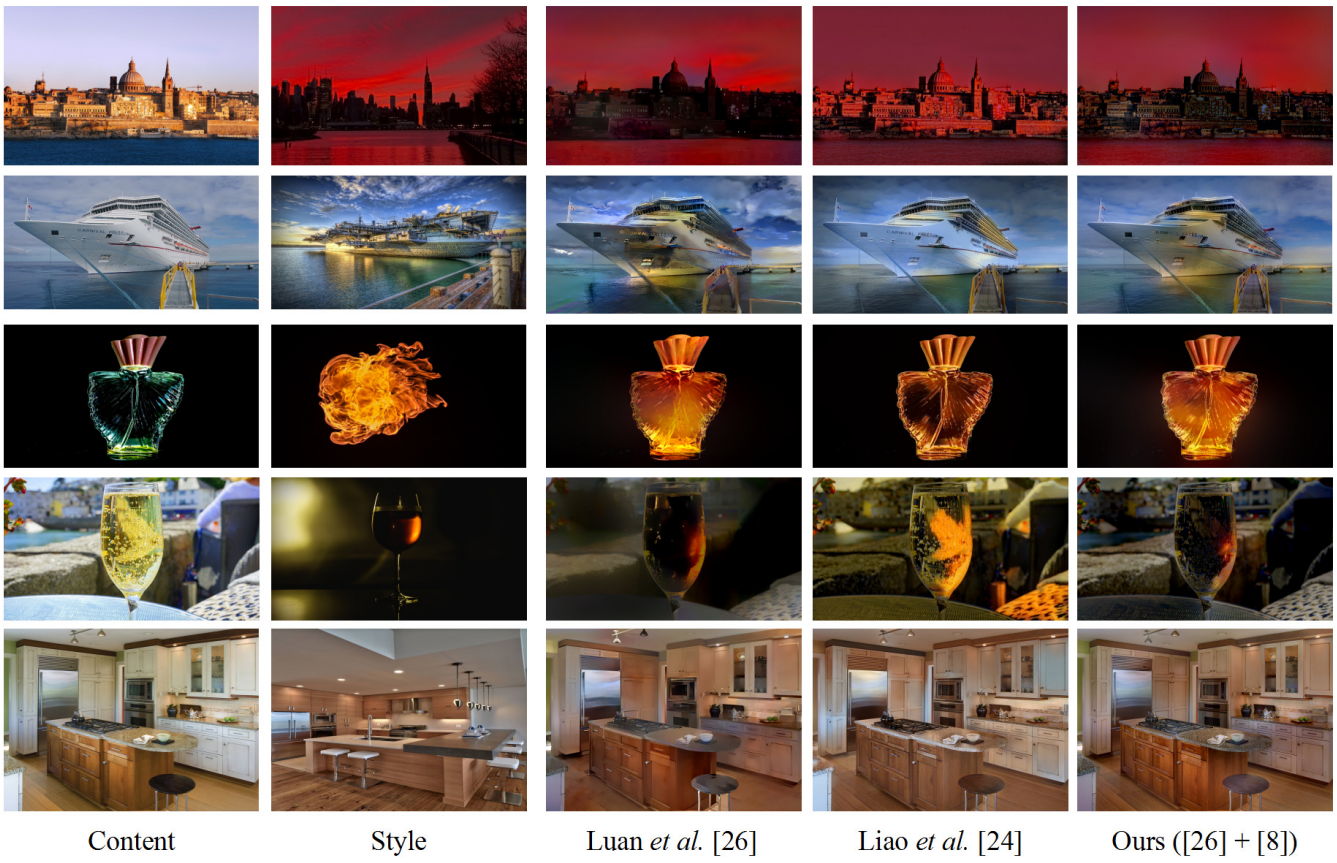

Figure 10: Comparison between state-of-the-art style transfer methods based on deep features [26], [24] and our refined results. All examples are from Luan *et al.* [26]. We recommend readers to view the electronic version.

[6] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*.

[7] Jacob R Gardner, Paul Upchurch, Matt J Kusner, Yixuan Li, Kilian Q Weinberger, Kavita Bala, and John E Hopcroft. 2015. Deep manifold traversal: Changing labels with convolutional features. *arXiv preprint arXiv:1511.06421* (2015).

[8] Eduardo SL Gastal and Manuel M Oliveira. 2011. Domain transform for edge-aware image and video processing. In *ACM Transactions on Graphics (ToG)*, Vol. 30. ACM, 69.

[9] Leon Gatys, Alexander S Ecker, and Matthias Bethge. 2015. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*. 262–270.

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2414–2423.

[12] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2016. Controlling perceptual factors in neural style transfer. *arXiv preprint arXiv:1611.07865* (2016).

[13] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. 2017. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830* (2017).

[14] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. 2011. Non-rigid dense correspondence with applications for image enhancement. *ACM transactions on graphics (TOG)* 30, 4 (2011), 70.

[15] Mingming He, Jing Liao, Lu Yuan, and Pedro V Sander. 2017. Neural Color Transfer between Images. *arXiv preprint arXiv:1710.00756* (2017).

[16] Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. *arXiv preprint arXiv:1703.06868* (2017).

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*. Springer, 694–711.

[18] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. 2014. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 149.

[20] Anat Levin, Dani Lischinski, and Yair Weiss. 2008. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 2 (2008), 228–242.

[21] Chuan Li and Michael Wand. 2016. Combining markov random fields and convolutional neural networks for image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2479–2486.

[22] Chuan Li and Michael Wand. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*. Springer, 702–716.

[23] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. 2017. Diversified texture synthesis with feed-forward networks. *arXiv preprint arXiv:1703.01664* (2017).

[24] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. 2017. Visual Attribute Transfer through Deep Image Analogy. *arXiv preprint arXiv:1705.01088* (2017).

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[26] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. 2017. Deep Photo Style Transfer. *arXiv preprint arXiv:1703.07511* (2017).

[27] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer*

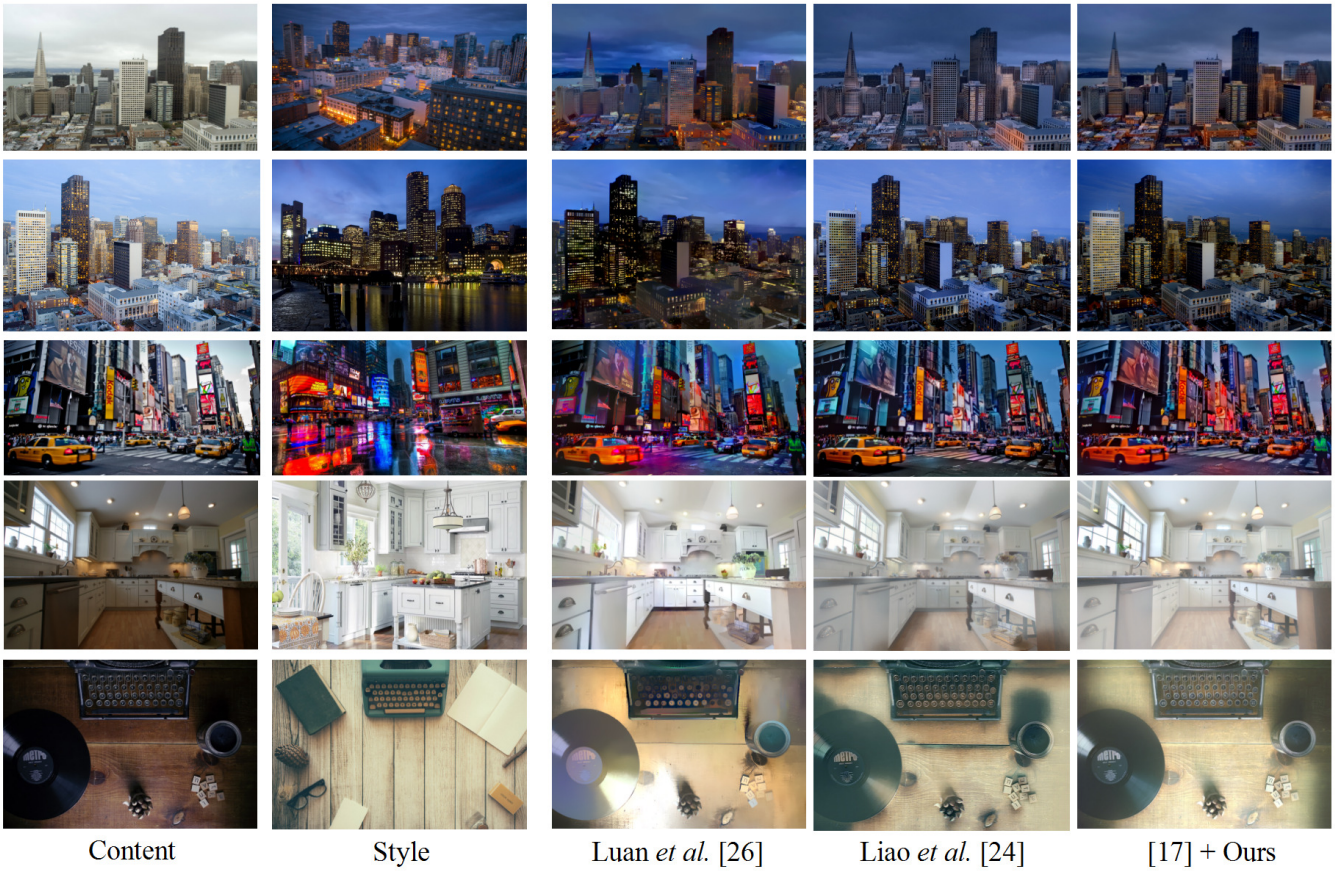

Figure 11: Comparison between state-of-the-art style transfer methods Luan *et al.* [26], Liao *et al.* [24] and ours ([17] + ours). All examples are from Luan *et al.* [26].

vision and pattern recognition. 5188–5196.
[28] Roey Mechrez, Eli Shechtman, and Lihi Zelnik-Manor. 2017. Photorealistic Style Transfer with Screened Poisson Equation. *arXiv preprint arXiv:1709.09828* (2017).
[29] Francois Pitie, Anil C Kokaram, and Rozenn Dahyot. 2005. N-dimensional probability density function transfer and its application to color transfer. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, Vol. 2. IEEE, 1434–1439.
[30] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. 2001. Color transfer between images. *IEEE Computer graphics and applications* 21, 5 (2001), 34–41.
[31] Falong Shen, Shuicheng Yan, and Gang Zeng. 2017. Meta Networks for Neural Style Transfer. *arXiv preprint arXiv:1709.04111* (2017).
[32] Yichang Shih, Sylvain Paris, Frédo Durand, and William T Freeman. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 200.
[33] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[34] Yu-Wing Tai, Jiaya Jia, and Chi-Keung Tang. 2005. Local color transfer via probabilistic segmentation by expectation-maximization. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1. IEEE, 747–754.
[35] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2017. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. *arXiv preprint arXiv:1701.02096* (2017).
[36] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. 2016. Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer. *arXiv preprint arXiv:1612.01895* (2016).
[37] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. 2002. Transferring color to greyscale images. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 277–280.
[38] Pierre Wilmot, Eric Risser, and Connelly Barnes. 2017. Stable and Controllable Neural Texture Synthesis and Style Transfer Using Histogram Losses. *arXiv preprint arXiv:1701.08893* (2017).
[39] Fuzhang Wu, Weiming Dong, Yan Kong, Xing Mei, Jean-Claude Paul, and Xiaopeng Zhang. 2013. Content-Based Colour Transfer. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 190–203.
[40] Jae-Doug Yoo, Min-Ki Park, Ji-Ho Cho, and Kwan H Lee. 2013. Local color transfer between images using dominant colors. *Journal of Electronic Imaging* 22, 3 (2013), 033003–033003.
[41] Hang Zhang and Kristin Dana. 2017. Multi-style Generative Network for Real-time Transfer. *arXiv preprint arXiv:1703.06953* (2017).
[42] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. 2015. Is L2 a Good Loss Function for Neural Networks for Image Processing? *ArXiv e-prints* 1511 (2015).

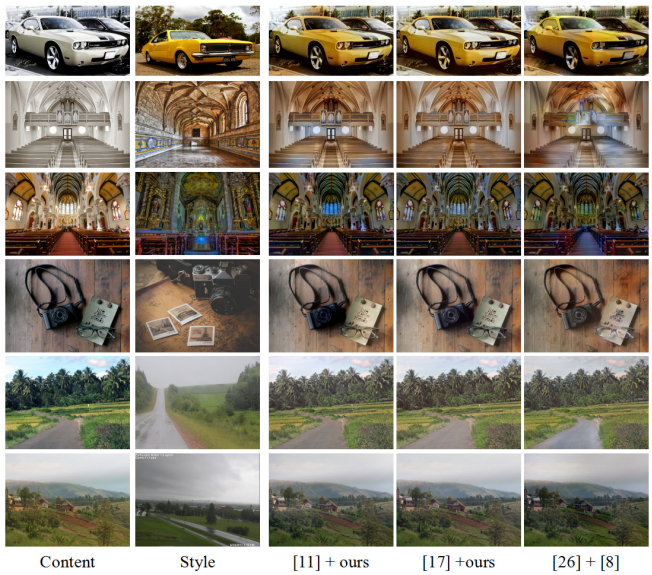

Figure 12: Comparison between our proposed methods [11]+ours, [17]+ours, and [26]+[8]. All examples are from Luan *et al.* [26]. We recommend readers to view the electronic version.

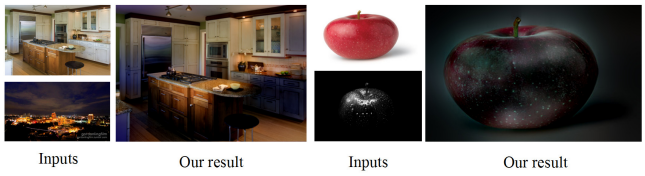

Figure 13: Some failure cases.

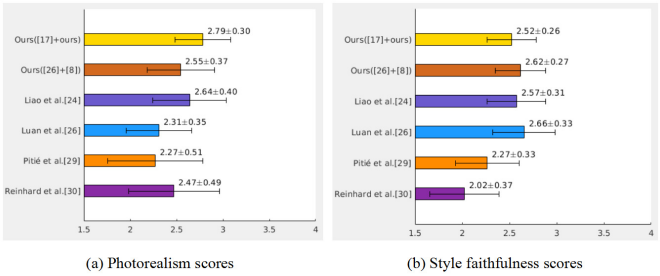

Figure 14: User study results for photorealism and style faithfulness.