Photo Style Transfer With Deep Learning

Ziyun Wang
New York University
zw2026@nyu.edu

Jiayang Wang
New York University
jw4149@nyu.edu

Abstract

Photo style transfer is the task of transferring the style of a reference photo onto a content photo while preserving the structural information of the content photo. Existing style transfer methods usually introduce artistic distortions when transferring styles, and cannot be applied to photographs directly. Some recent works propose different techniques to avoid content mismatching and unrealistic distortions, and make the results more photographic. In this paper, four different techniques are studied, including a photorealism regularization, an L1-norm based similarity loss, a new style loss with semantic segmentation, and a filter-based post process. We also enhance the semantic style loss with soft semantic segmentation. Finally, we perform thorough qualitative evaluations, and propose a combined model, which obtains finer style transfer results. The code of this project can be found at `https://github.com/Billijk/Deep-Photo-Style-Transfer-PyTorch`.

1. Introduction

The task of applying the style, such as weather or illumination, of one photo to another photo is meaningful but difficult in image processing. One example is to transfer the dark night of city A in one picture onto the landscape of city B in another picture. This kind of image processing vastly appears in scenarios of advertising, photo processing, movie making, *et al*. However, currently this task still mainly rely on the effort of human experts with domain knowledge, such as chromatology and aesthetics. This task has been studied in the field of Computer Vision and Computer Graphics since years ago, and a lot of methods transferring color, texture, or style were proposed. With the advances in deep learning during recent years, a series of artistic style transfer methods based on Convolutional Neural Networks(CNN) have achieved great success[2, 4, 6, 12], starting from the work of Gatys *et al*. [2], which measures the distance of pictures using high-level features captured by pretrained CNN feature extractors, and minimizes distance of the output image to content image and style image.

While these methods generate visually pleasing pictures, especially with artistic styles, they usually fail to maintain the photorealism of the content picture, because of content-mismatching and distortions, which is shown in figure 1. Thus they cannot directly apply to style transfer task in photographs. To solve this limitation, Luan *et al*. [7] proposed a deep photo style trasfer method, which includes a photorealism regularization term based on locally affine errors, and a semantic segmentation based style loss term which limits the style transfer on same semantic objects to prevent content-mismatching. The content structures are preserved in many cases, but details of edges are usually missed due to inaccurate semantic segmentation. Errors of segmentation will also inevitably being propogated to the final results. Mechrez *et al*. [8] proposed Screened Poisson Equation (SPE) as a replacement for the photorealism regularization term of Luan *et al*.'s, which tries to correct the gradient on the output image to be more similar to the content image. Wang *et al*. [15] proposed a simple yet effective post processing technique to replace the regularization terms. They use a recursion filter which does not interfere with the back propagation, computes more efficiently, and yields better results then the regularization term. Their method also uses a similarity loss based on L1-Norm to replace semantic segmentation, but it cannot solve the content mismatch problem perfectly.

In this paper, we mainly focus on the work of Luan *et al*. and Wang *et al*., and study the photorealism regularization, the semantic augmented style loss, the similarity loss, and the post processing based on recursive filter. To alleviate the problem caused by semantic segmentation, we improve it with the idea of soft segmentation, which allows one pixel on the content image to take styles from different semantic objects in the style image, by letting pixels of both images to have multiple semantic labels at the same time. We propose a new method with some of the techniques combined, and perform large scale of experiments. From the experiments, our proposed method can generate better results in many cases. The soft semantic segmentation introduces possibilities of correcting segmentation errors, and also provide more natural transitions of colors, especially at the edges of

Figure 1: Traditional neural style transfer algorithms often introduce artistic distortions (such as the strokes on the arch) and content-mismatching (the arch should be white and the trees on the right should not be black). Our model better preserves the structures of the content, and transfer the corresponding styles correctly.

two semantic objects. The other techniques are also proved useful in our experiments.

To sum up, our contributions are two-folds.

1. We propose an improved style loss based on soft semantic segmentation, which alleviates many problems caused by traditional semantic segmentation.
2. We implemented and evaluated many techniques proposed by previous works, and designed a combined model which generates better style transfer results in many cases.

2. Related Works

Traditional style transfer algorithms can be roughly divided into two categories, namely global style transfer and local style transfer. Global style transfer methods apply a spatial-invariant transformation to the image, such as a global color shift [11]. These methods can handle simple styles easily, but are limited in complex situations. On the other hand, local style transfers take the spatial information into consideration, so they can handle more sophisticated styles such as weather or time-of-day change. However, these methods require user inputs [16] or image segmentation [14] as spatial correspondence guidance, and may introduce inaccurate transfers due to spatial mismatch.

Neural style transfer [2, 4] proposed in recent years is a special category of local style transfers, which are very powerful with the feature extraction ability of convolutional neural networks. These methods generate quite promising results especially with artistic styled reference images. Following these work, Luan *et al*. [7] first introduce the task and challenges of style transfer on photographs. Mechrez *et al*. [8], Wang *et al*. [15] also propose many techniques to preserve the realism of photographs while transferring the style. Our work in this paper directly follow the work of Luan *et al*. and Wang *et al*.

3. Methods

In this section, we will introduce the basic neural style transfer model, and several improvements proposed recently towards generating more realistic results, including the soft semantic segmentation technique we proposed.

3.1. Baseline Model

The baseline model is the one proposed by Gatys *et al*. [2], where the style image S is transformed onto the input image I to produce an output image O that preserves the structures in I, but immersed in the context of S. The idea of this approach is to let a pre-trained Deep Convolutional Neural Network detect and extract the high-level features of the images, and define a content loss and a style loss out of these feature representations learned by the neural network. The losses are designed to be measurements for distance of images, and the task is eventually accomplished by minimizing a linear combination of the two defined losses, in other words, minimizing the distance.

The objective loss function would be

$$\mathcal{L} = \alpha \sum_{l \in L_c} \mathcal{L}_c^l + \beta \sum_{l \in L_s} \mathcal{L}_s^l \tag{1}$$

where L_c and L_s are two sets containing layers used for computing content loss and style loss, respectively. α and β control the weights of the two types of losses. Content loss and style loss are defined as

$$\mathcal{L}_c^l = \frac{1}{2N_l D_l} \sum_{ij} (F_l[O] - F_l[I])_{ij}^2 \tag{2}$$

$$\mathcal{L}_s^l = \frac{1}{2N_l^2} \sum_{ij} (G_l[O] - G_l[S])_{ij}^2 \tag{3}$$

where F_l is the feature matrix at the lth layer, and $G_l = F_l F_l^T$ is the Gram matrix associated the filters.

An overview of the model is shown below. [2]

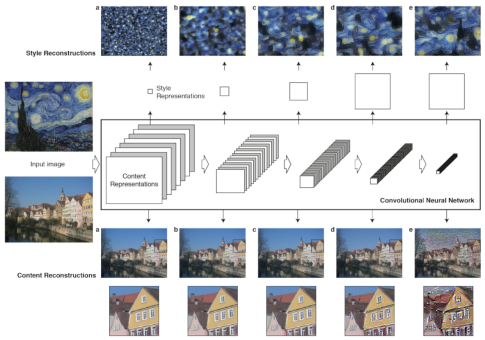

Figure 2: Structure of the Baseline Model [2]

3.2. Photorealism Regularization

In Luan *et al*. [7], the authors proposed to add a penalty term to "penalize" distortions. Under the hood, the strategy is to have a transform that is locally affine for each patch in the color space, but the transforms can be different from patch to patch. In this way, the edges of image elements can be located at the same place in all channels while allowing spatial variations.

Such a transform can be formulated as:

$$\mathcal{L}_m = \sum_{c=1}^{3} V_c[O]^T M_I V_c[O] \tag{4}$$

and its derivative with respect to the output image can be formulated as

$$\frac{d\mathcal{L}_m}{dV_c[O]} = 2M_I V_c[O] \tag{5}$$

where M_I is the matting Laplacian [5] that only depends on the input image, and $V_c[O]$ is the vectorized (flattened) version of the output image. This transformation will be added to the objective function we are seeking to minimize as the penalty term.

3.3. Semantic Augmented Style Loss

It was proposed in Luan *et al*. [7] that one important issue with the baseline algorithm is the fact that the gram matrix was computed over the entire style reference image. The direct result of this approach is the mis-alignment of different elements in the picture – such as the color of the buildings affecting the color of the sky and vice versa. In order to address this issue, segmentation masks were generated for both content and style images for a set of common labels and were used as extra channels in addition to RGB pixel values. The style loss was then updated as follows.

$$\mathcal{L}^l_{s+} = \sum_{c=1}^{C} \frac{1}{2N_{l,c}^2} \sum_{ij} (G_{l,c}[O] - G_{l,c}[S])^2_{ij} \tag{6}$$

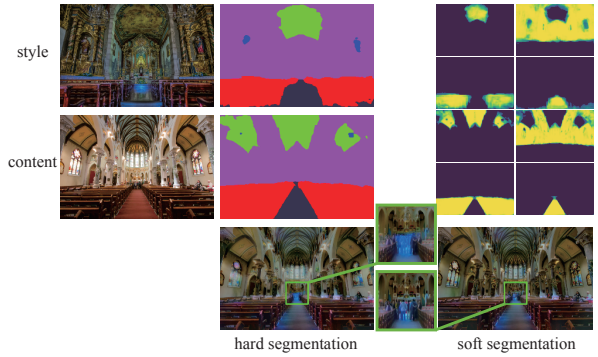

Figure 3: Differences between hard and soft segmentation. The semantic segmentation results are also shown. For hard segmentation, different colors show different semantic labels. For soft segmentation, different colors show differences in possibilities, with deeper color shows smaller possibility. Only 4 main semantic categories are shown here for illustration.

with

$$F_{l,c}[O] = F_l[O]M_{l,c}[I]\,,\; F_{l,c}[S] = F_l[S]M_{l,c}[S] \tag{7}$$

where C is the number of channels, or masks, we have, $M_{l,c}$ is the segmentation mask in layer l corresponding to label c, and $G_{l,c}$ is, as before, the Gram matrix for its corresponding $F_{l,c}$.

Such modification effectively limits style transfer only to contents with the same semantic label, and successfully prevents the "spillovers" in many situations. However, if the semantic segmentation is inaccurate, the results will be even worse since it is unlikely for style to be applied on the correct contents.

To further address this issue, we propose soft semantic segmentation in this paper. Instead of having a hard segmentation where each pixel has only one semantic label, we calculate for each pixel a distribution of its semantic labels. In this case the style loss is the same as the one defined in equation 6 and 7, with $M_{l,c}[I]$ and $M_{l,c}[S]$ being "soft" masks, which can take contiguous value between $[0, 1]$. The soft segmentation can be easily implemented with most semantic segmentation models whose last layer is a softmax activation. Without taking an argmax, we directly use the values after softmax as segmentation results. In practice, we set a cutoff threshold θ_s to ignore labels with small possibilities to reduce computation. Figure 3 shows an example comparing hard and soft semantic segmentation. With hard segmentation, people in the content image are falsely labeled as ground, and they get the blue color after style transfer. On the other hand, they are less influenced by the blueish style of ground with soft segmentation.

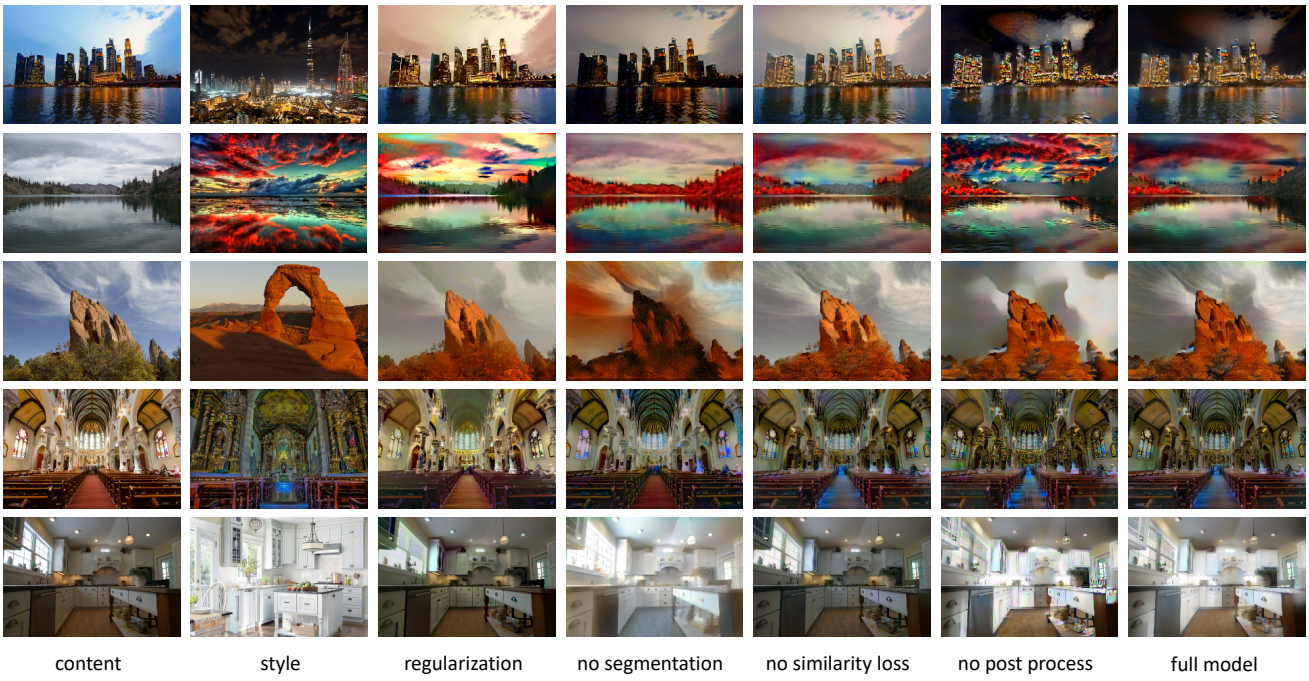

Figure 4: Ablation study results. Input content images and style images are shown in the left two columns, the rest are results from 5 different model variants. Examples are from Luan *et al*. [7].

3.4. L1-Norm Based Similarity Loss

According to Zhao *et al*. [17], for tasks such as image construction, L1-norm loss outperforms L2-norm loss for the same CNN architechture. So Wang *et al*. proposed a similarity loss function in their work [15], which computes the mean absolute error of the feature maps. The similarity loss for layer l is

$$\mathcal{L}_{sim}^{l} = \frac{1}{2N_l D_l} \sum_{ij} \left| F_l[O] - F_l[I] \right|_{ij} \tag{8}$$

3.5. Recursive Filter

In the work by Wang *et al*. [15], they employed Recursive Filter [1], a 2D edge preserving filter for domain transformation, to refine the output image O with guidance of the input content image I. The refined image is defined as

$$O_p = (I - RF(I, \sigma_s, \sigma_r, I)) + RF(O, \sigma_s, \sigma_r, I) \tag{9}$$

where RF denotes Recursive Filtering, which takes four arguments, image to be filtered, filter spacial standard deviation σ_s, filter range standard deviation σ_r, and the image for joint filtering (in our case is I).

3.6. Our approach

We combine some techniques with the baseline model to construct our final model, including soft semantic segmentation based style loss, similarity loss, and post processing. The total loss function in our model is a combination of three parts

$$\mathcal{L}_{total} = \alpha \sum_{l \in L_c} \mathcal{L}_c^l + \beta \sum_{l \in L_{s+}} \mathcal{L}_{s+}^l + \gamma \sum_{l \in L_{sim}} \mathcal{L}_{sim}^l \tag{10}$$

where α, β, γ are weights of different types of losses.

We do not adopt photorealism regularization, because it tries to solve the same issue as post processing, but it is much more expensive in computation. Besides, according to our experiments, adding a term in loss function has side effect in the training process and sometimes produce undesired results. We will discuss more on this in section 5.1.

4. Implementation Details

We re-implement all the techniques mentioned in section 3 in PyTorch [9]. We utilized a pre-trained VGG-19 [13] network as the feature extractor used in neural style transfer. Following Alexis Jacq's implementation available from the PyTorch tutorials [3], our L_c contains the feature maps produced by the fourth convolutional layer, L_{s+} contains the feature maps produced by the first five convolutional layers, and L_{sim} contains the feature maps from the first three convolutional layers. For all images, initial learning rate is set to 0.1, the content weight α is set to 1, the style weight β is set to 10^7, and the similarity weight γ is set to 3. The maximum number of iterations is 500. However, we

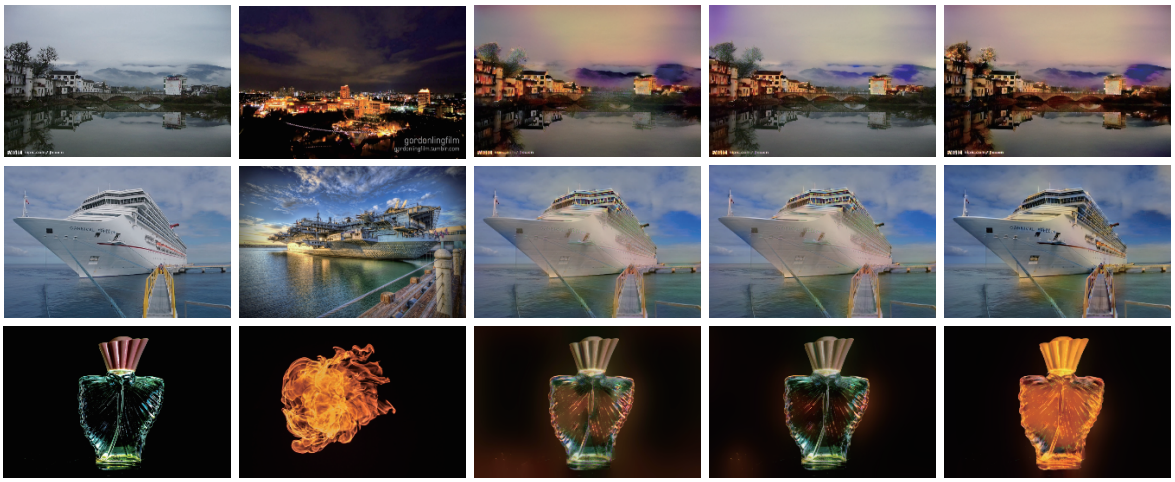

Figure 5: Comparison of results generated by models with hard segmentation, soft segmentation, and gold segmentation provided by Luan *et al*. [7]. We recommend readers to view the higher resolution images in supplemental materials to compare details.

observe that sometimes for some images the loss will suddenly increased to a huge value during optimization. Thus we decrease the number of iterations for these images to prevent this phenomena.

We use the pre-trained models provided by Zhou *et al*. [18] to generate labeled masks for both content and style images. To prevent discrepancy between the genres of labels in both pictures, we constrain the labels of content image only to the ones present in the style image. We also merged similar labels such as 'lake' and 'river', as suggested by the authors in the original paper [7].

For calculating Matting Laplacian [5] and perform Recursive Filtering [1], we directly use the MATLAB codes provided by the authors of these two papers. For Recursive Filtering, σ_s is set to 60, and σ_r is set to 100.

5. Experiments

To compare the performance of different models, in this section, we will conduct experiments and perform qualitative comparisons on the results.

5.1. Ablation Study

We first compare the results of our full model with the results obtained by removing one of the modules, to illustrate the functionality of each module we adopt. The results are shown in figure 4. In this comparison, we show the results of 5 different variants of our model. The full model uses the soft semantic segmentation module, similarity loss, and post processing. The regularization model replaces the post processing with the photorealism regularization introduced in section 3.2. The other three models remove the soft semantic segmentation module, similarity loss, and post processing, respectively.

The photorealism regularization effectively preserves the structure of input image, but as a plugin term into the loss function, it also has an impact on the optimization direction. As a result, the generated image may not reflect the style perfectly. Besides, calculating the matting Laplacian used in this regularization term is extremely slow even for image with low resolutions. It also slows down the backpropagation since it requires more computation for each step.

As expected, semantic segmentation plays an important role in solving the content-mismatch issue. In the second and the third example in figure 4, the model without segmentation mistakenly applies the red color to different regions, while the full model provides correct mapping of styles. Similarity loss also helps to generate results which better captures the style information. A possible reason is that it provide more information to correct the gradients than with mean square error alone, and helps the back propagation to converge more quickly. Post processing step is able to effectively remove the distortions. Figure 7 provide more details to compare the images before and after post processing, where images after post processing show much clearer structure than the images before post processing.

5.2. Impact of Semantic Segmentation

As mentioned before, one drawback of Luan *et al*.'s work is that the error of semantic segmentation will misguide the content-matching seriously. We propose soft semantic segmentation to cope with this issue in this paper.

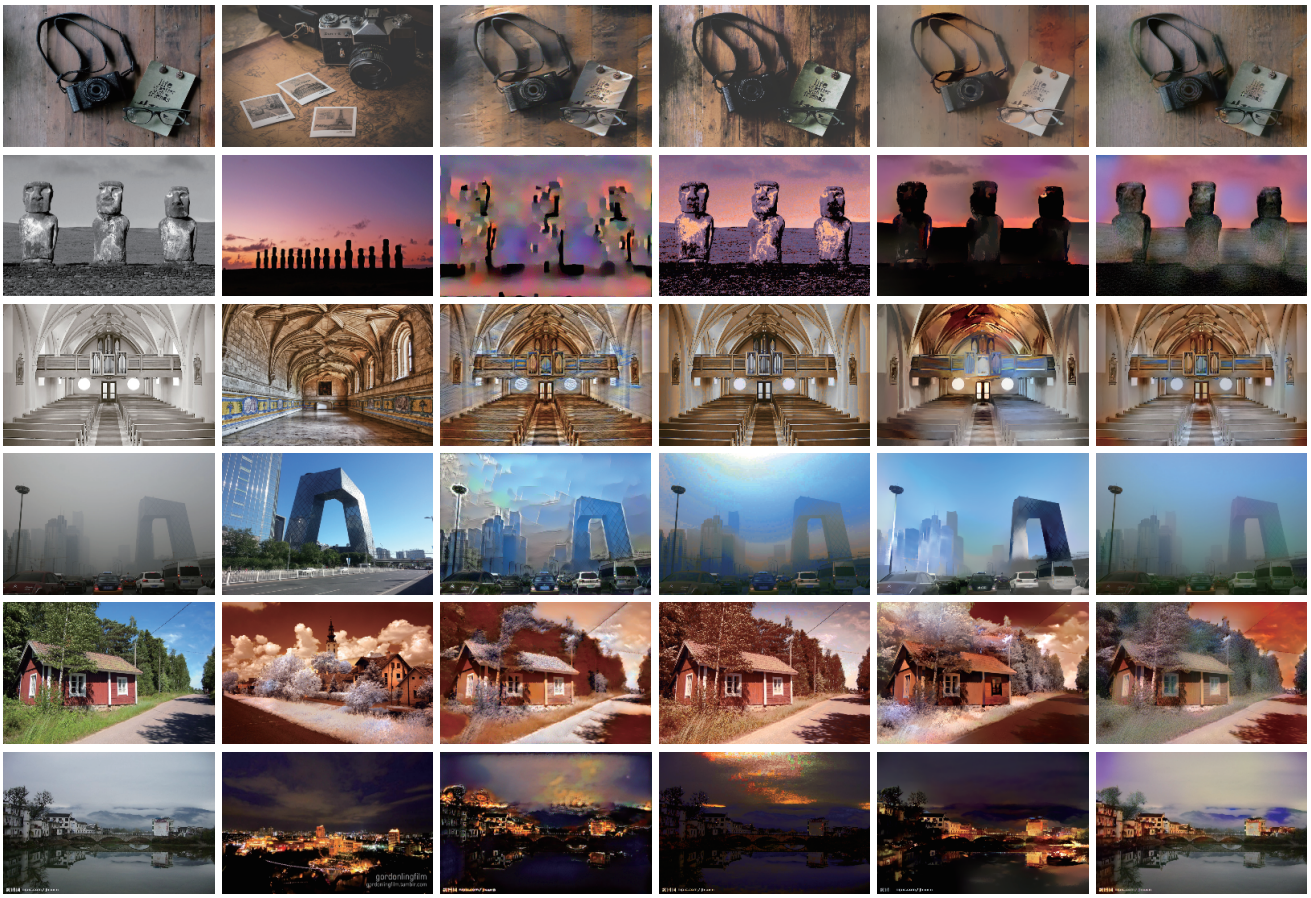

Figure 6: Comparison of our method against Gatys *et al*. [2], Pitié *et al*. [10], and Luan *et al*. [7].

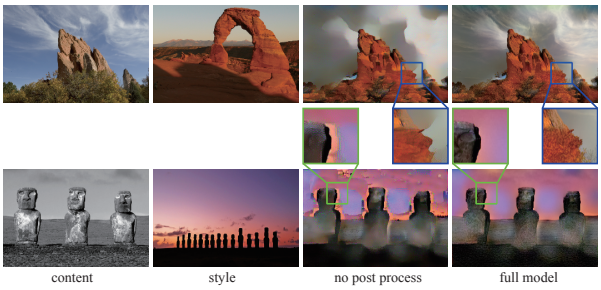

Figure 7: Examples showing how post processing step removes the distortions and preserves the structure of the input image.

Figure 5 shows how segmentation error may influence the generated picture.

Soft segmentation can alleviate the false style transfer due to inaccurate segmentation in some situations. In the first row of figure 5, the right part of the bridge has different semantic label as the left part due to an error. With soft segmentation, the color of the bridge does not have a sudden change as the case with hard segmentation. Besides, hard segmentation may produce unnatural style split around edges, such as the trees and sky on the left part of the first image. Soft segmentation can generate a more natural transition between two different styles.

However, soft segmentation cannot completely prevent the content mismatch. In the second example of figure 5, hard and soft segmentation get almost the same result, while the gold segmentation helps to generate a much better output. We also illustrate here that with user provided segmentation, we can transfer styles from totally different objects. In the third example, we successfully transfer the style of fire onto a glass bottle with proper segmentation masks. This technique can be used to produce some interesting photograph editing with special effects.

5.3. Comparison to Previous Work

We also compare our model with previous works, including the baseline artistic style transfer model by Gatys *et al*.

Figure 8: Some bad results generated by our method. The segmentation results are also provided, along with their semantic labels. For each label, top one is the semantic mask for content image, and below one is the mask for style image.

[2], a well-known global color transfer method proposed by Pitié *et al*. [10], and the deep photo style transfer model by Luan *et al*. [7]. The results are provided in figure 6.

The method of Gatys *et al*. inevitably introduces many artistic distortions, and erases important structure details in most cases, which is the challenge photographic style transfer wishes to solve. On the other hand, the global transfer by Pitié *et al*. preserves the structure in most cases, but it can only transfer a universal color distribution, which ignores spacial information and the whole image has a homogeneous style with content mismatching. The photo style transfer method by Luan *et al*. tries to solve both issues and achieves promising results in many cases. It also has many limitations. In the examples in figure 6, some details are blurred in the results of this method, like the stone figures in the second example, and the buildings in the fourth example. It may suffer from the segmentation issue that some regions in the content image do not have enough style information for reference. And its photorealism regularization is less effective than post processing technique.

The method of Luan *et al*. generates better results in some cases (example 5 and 6 in figure 6) than our method, especially the style of their results are more similar to the style image. We believe this is mostly due to a parameter issue. We do not tune the parameter on each individual image, and choose the same set of parameters for all inputs except the number of training iterations. A careful selection of parameters with respect to the input will possibly produce a better result.

5.4. Failure Case Study

Besides the parameter issue, we also discover some other sources of generating bad results in some cases. Figure 8 provides two examples. In the first example, the style of house and other objects (trees, sky, etc.) are very different in the style image. With soft segmentation, those other objects may falsely be transferred with style of house, which produce a weird combined style. In the second example, the styles are transferred almost correctly, but it is hard to precisely convert the green trees into white ones with correct details. Another interesting thing is that the reflection of trees in the water actually has both semantic label of water and trees, so it is potentially able to take both styles. However, the result is not very satisfactory. One future direction of this work is to explore the potential of using soft semantic segmentation for style overlap.

6. Conclusion

In this paper, we studied the task of generating photorealistic style transferred images. We explored many previous methods on solving this task, and proposed soft semantic segmentation, which alleviates an serious drawback of an existing method. Through the massive evaluations, we proved the effectiveness of our method. However, we also found cases where our method produced unsatisfactory results. How to generate natural results in complicated scenarios is still a challenge that remains to be solved. Another future direction is to explore how soft segmentation can be better used for regions with overlapped semantic types, as discussed in section 5.4.

References

[1] E. S. Gastal and M. M. Oliveira. Domain transform for edge-aware image and video processing. In *ACM Transactions on Graphics (ToG)*, volume 30, page 69. ACM, 2011.

[2] L. A. Gatys, A. S. Ecker, and M. 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] A. Jacq. Neural tranfer using pytorch. `https://pytorch.org/tutorials/advanced/neural_style_tutorial.html`, 2017.

[4] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[5] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2008.

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

[7] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 4990–4998, 2017.

[8] R. Mechrez, E. Shechtman, and L. Zelnik-Manor. Photorealistic style transfer with screened poisson equation. *arXiv preprint arXiv:1709.09828*, 2017.

[9] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[10] F. Pitie, A. C. Kokaram, and R. Dahyot. N-dimensional probablility density function transfer and its application to colour transfer. In *null*, pages 1434–1439. IEEE, 2005.

[11] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[12] A. Selim, M. Elgharib, and L. Doyle. Painting style transfer for head portraits using convolutional neural networks. *ACM Transactions on Graphics (ToG)*, 35(4):129, 2016.

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Y.-W. Tai, J. Jia, and C.-K. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 747–754. IEEE, 2005.

[15] L. Wang, N. Xiang, X. Yang, and J. Zhang. Fast photographic style transfer based on convolutional neural networks. In *Proceedings of Computer Graphics International 2018*, pages 67–76. ACM, 2018.

[16] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring color to greyscale images. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 277–280. ACM, 2002.

[17] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Is l2 a good loss function for neural networks for image processing. *ArXiv e-prints*, 1511:8, 2015.

[18] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision*, 2018.