Understanding and Comparison of Photorealistic Image Stylization

Zhehan Shi [*1 2] **Mei Chen** [*1 2] **Zixiang Pei** [*1 2]

Abstract

Photorealistic image stylization concerns transferring style of a reference photo to a content photo with the constraint that the stylized photo should remain photorealistic. Whitening and coloring transforms (WCT), a VGG-19 based network architecture that utilized multi-level stylization, is one of the avant-garde methods in generating high quality stylized images. We focus on a variant of WCT called PhotoWCT which specializes in photorealistic style transfer, and experiment on how training on different model architecture and layers affect how well the stylized image preserves the original content structure. Our main qualitative metric will be the structural similarity index measure (SSIM).

1. Introduction

Style transfer is a heated task in the field of modern computer vision. Given a target photo and a reference photo, the model should synthesize a new image that preserves the original scenes but adds in the reference photo's styles. A famous such algorithm would be WCT, which utilizes whitening and coloring transform to directly match content feature statistics to those of a style image in the deep feature space (Li et al., 2017). Photorealistic style transfer, on the other hand, requires preserving the details of an image without distortion. The PhotoWCT network architecture proposed in Li et al. (2018) is a state-of-the-art technique for enhancing photorealism and transferring content image styles.

PhotoWCT improves upon WCT in that it replaces the upsampling components of the VGG decoder with unpooling. In addition, it utilizes perceptual loss (Johnson et al., 2016) and uses a sum of perceptual loss and L2 reconstruction loss to train its decoder. Finally, it proposes a photorealistic smoothing scheme based on matting affinity of photos (Levin et al., 2007) to solve the problem that semantically similar regions are often stylized inconsistently. In this paper, we mainly achieve three goals: firstly, we reconstruct the training code for PhotoWCT and probe how individual layers contribute differently in terms of preserving the original content and transferring colors based on the style image; secondly, we observe that the original training and inference procedure is time-consuming, so we propose and experiment on novel training and inference procedure and demonstrate whether we can achieve comparable results using a more efficient procedure; finally, we conduct a mini-investigation to compare photorealistic smoothing with guided-filter-smoothing. Figure 1 shows an example of photorealistic style transfer under our implementation. The implementation can be found at `https://github.com/cyberzzhhss/neuralStyleTransfer` or `https://github.com/Joepei/cv_Final`.

[*]Equal contribution [1]New York University [2]Center for Data Science. Correspondence to: Zhehan Shi <zs1113@nyu.edu>, Mei Chen <mc8895@nyu.edu>, Zixiang Pei <zp2123@nyu.edu>.

2. Related Work

Neural Style Transfer. Style transfer is a long-standing topic, and since Gatys et al. (2015; 2016) presented how to utilize the Convolutional Neural network to reproduce painting styles on natural image, there were numerous pieces of literature presenting different synthetic stylized images based on neural networks. Jing et al. (2019) summarized that most of the neural style transfer algorithms utilized a combination of visual texture modelling method with image reconstruction to achieve image style transfer, examples includes Gatys et al. (2016); Heeger & Bergen (1995); Ulyanov et al. (2016), which used online image optimizations. However, those methods were computationally expensive and they could not restore photorealistic images because of their abstraction in features.

Photorealistic Image Style Transfer. Photorealistic style transfer aims to build upon current style transfer while eliminating the distortions and preserving the original structural details. The earliest literature in this field that used neural network is Luan et al. (2017). It proposed deep photo style transfer that augmented the algorithm in Champandard (2016) and added a penalizing term for non-photorealistic distortions. Under the same idea, Mechrez et al. (2017) proposed a normalization term called screened Poisson equa-

(a) Content Image (b) Style image (c) Style Transfer w/o Smoothing (d) Style Transfer with Smoothing

Figure 1. An example of our implementation of style transfer

tion that was more computationally efficient. Li et al. (2017; 2018); Yoo et al. (2019) proposed a fundamentally different approach from previous feed-forward networks; they used a learning-free scheme to efficiently extract arbitrary style from the example and reconstructed the image without further fine-tuning, which will be further discussed in section 3.

3. Methodology

In PhotoWCT, there are two steps to build the model:

1. It first separately trains mirror-structured decoders, to reconstruct images from each layer of the pretrained VGG-19 based encoder.
2. To stylize an image, PhotoWCT extracts features from its content and style using an encoder, combines features using WCT functions and reconstructs them back using the trained decoder.

3.1. WCT Transformation

WCT refers to the whitening and coloring transformation (equation 1) between each layers of encoder and decoder to achieve transformed feature map H_{CS}:

1. whitening transformation, which is essentially a zero-phase whitening (Bell & Sejnowski, 1997) on the normalized content image's feature vectors H_C, making pixels de-correlated and whitened;
2. coloring transformation, which is a reverse operation to the previous whitening operation that multiplies back the whitened feature map from content image with the style images' features (P_S).

The goal of WCT is to directly transform the content VGG feature maps at a certain layer to match the covariance matrix of the style feature map.

$$H_{CS} = P_S P_C H_C \tag{1}$$

, where

$$P_C = E_C \Lambda_C^{-\frac{1}{2}} E_C^T \tag{2}$$

$$P_S = E_S \Lambda_S^{-\frac{1}{2}} E_S^T \tag{3}$$

3.2. Variations of Training/Inference

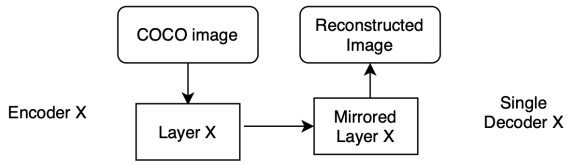
The original PhotoWCT model was reliant to the VGG-19 network to train the reconstructed decoder. We are curious how much improvement each layer gives to the stylized image. Thus, we suggest alternate methods for training decoders and performing style transfer.

Decoder Training

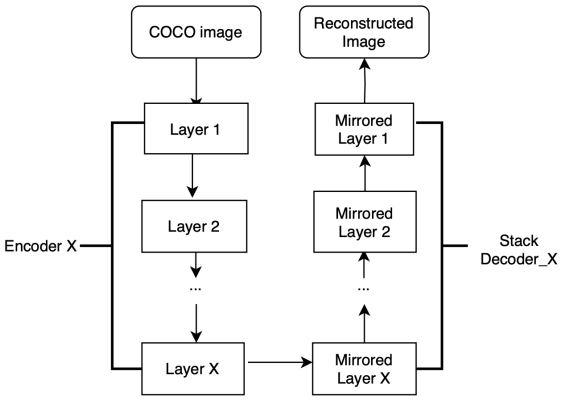
As in figure 2b, PhotoWCT's training utilized multiple layers of VGG-19 as the encoder to extract the feature, and constructed the corresponding decoder that has a mirrored architecture to the encoder to reconstruct the decoder, which we refer to as a stacked decoder. We would like to know whether a more shallow layer would be able to achieve comparable results effectively, and thus we further train a single decoder, as figure 2a shows, that only uses one level of VGG-19 model to extract the features from the image and try to reconstruct it with a single layer decoder. We use the same loss function (feature loss, reconstruction loss proposed in Dosovitskiy & Brox (2016)) as that used in WCT during the training.

Stylization Inferences

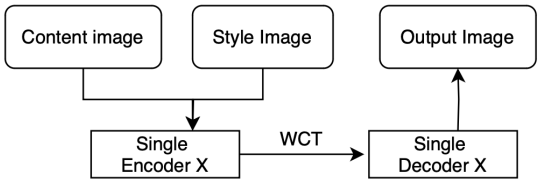
Figure 2d displays the "Multi-level coarse-to-fine stylization" method introduced by WCT and PhotoWCT. This pipeline utilizes iterative feature extraction and reconstruction in each layer, and feeds the resulting image to the next encoder layer. The authors argue that with a deeper feature space can more effectively capture style characteristics from low to high levels during the style transfer process. To validate this claim, we re-implement both the multi-inference and single inference (Figure 2c) approaches and measure their efficacy by displaying and quantitatively calculating the structural similarities for all the intermediate results.



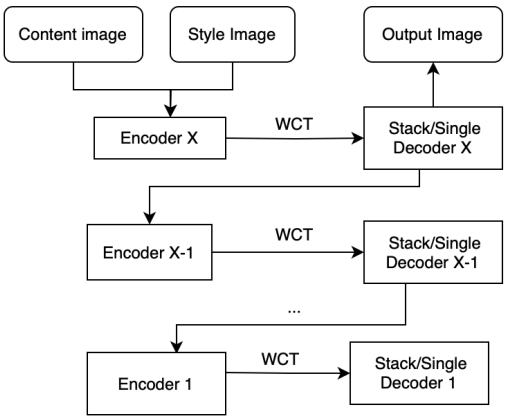
(a) Single Layer Training;
(each layer stands for one layer in VGG-19 model)



(b) Multiple Layer Training (Stacked Decoder)



(c) Single Inference



(d) Multi-Inference

Figure 2. Model Architecture for different training/inference processes, where WCT refers to the transformation of whitening and coloring. There are three models to compare: 1. single inference with single decoder is called the single decoder model; 2. Multi-Inference with single decoder is called the Accumulative Decoder Model; 3. Multi-Inference with Stacked Decoder is called the Stack Decoder model.

4. Experiments

4.1. Implementation Details

Because we are not able to find the training code from Li et al. (2018), we re-implement the training process based on the description from the paper. We choose VGG-19 network pre-trained on ImageNet to be our encoders' weight. Following the suggestions in (Li et al., 2017), we trained all decoders separately with frozen encoders to reconstruct images on the Microsoft COCO dataset (Lin et al., 2014) with randomized initial weight. We train 10 epochs in both single-layer and multiple-layer training paradigm.

Additionally, we also adapt implementations that can incorporate information from semantic label maps in style transfer when there exists a segmentation. We adapt the same segmentation mapping method from (Yoo et al., 2019; Luan et al., 2017). We conduct quantitative measurements on 60 images in Luan et al. (2017) that have segmentation maps, and the results will be present in the following sections.

4.2. Visual comparison

Figure 1 illustrates our approach of retaining all content image details while transferring the color palette from the style image. However, PhotoWCT exhibits sporadic artifacts (Figure 1.c) despite preserving content image details and structure. We use additional smoothing techniques proposed by PhotoWCT to achieve the desired visual quality as discussed in Section 5.

Figure 3 provides a more comprehensive comparison of various training and inference paradigms. The first row presents the raw content and style image, and the other rows contain stylized images from different layers with different training/inference paradigms.

(a) Content Image

(b) Style Image

(c) Single Decoder 1

(d) Single Decoder 2

(e) Single Decoder 3

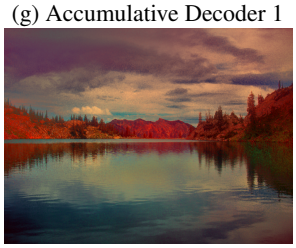
(f) Single Decoder 4

(g) Accumulative Decoder 1

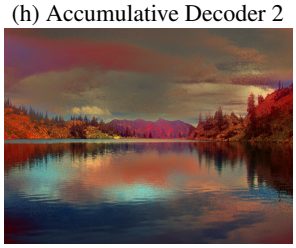
(h) Accumulative Decoder 2

(i) Accumulative Decoder 3

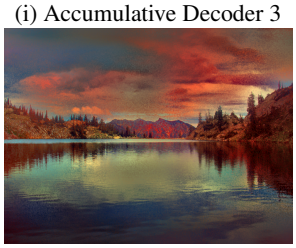
(j) Accumulative Decoder 4



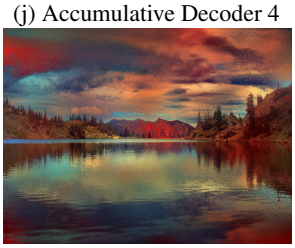
(k) Stacked Decoder 1



(l) Stacked Decoder 2



(m) Stacked Decoder 3

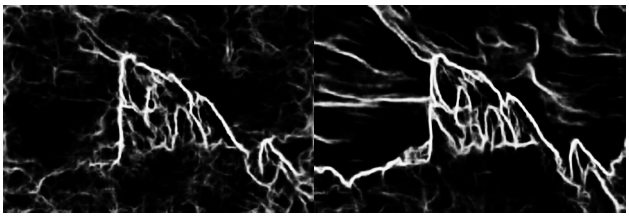


(n) Stacked Decoder 4

Figure 3. Comparison Between Accumulative, Stack, and Single Layer Style Transfer

4.3. Metrics

In addition to visual evaluations, we adapt the proposed method used in Yoo et al. (2019) and use Structural Similarity Index Metric (SSIM) to confirm how much details are preserved in transformed images. See Figure 4 as a demonstration: we utilize the state-of-the-art edge detection from Su et al. (2021) to first generate edge maps from both raw content images and from transferred images, and then calculate the structural similarity to have a quantitative measure.



(a) Content Image edge (b) style transferred-image edge

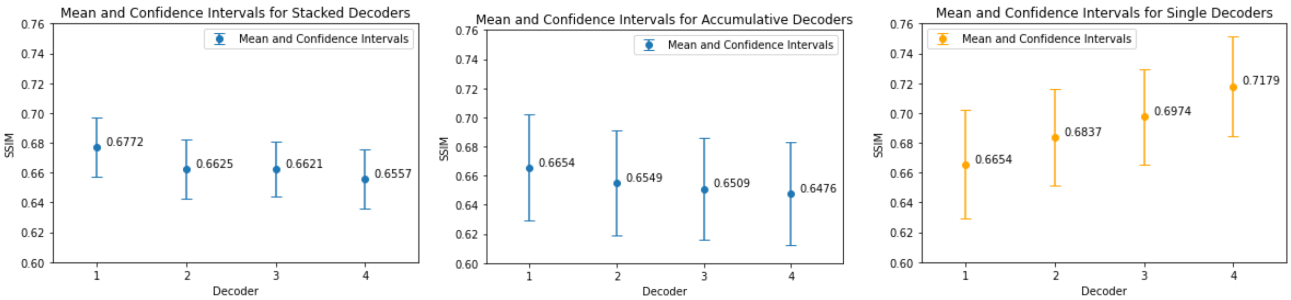
Figure 4. edge comparison example with SSIM of 0.65

5. Results and Conclusion

5.1. Training Results and Discussions

Comparison between Different Models

Using the metrics defined in section 4.3, we demonstrate the stylization results for stack and accumulative decoders in Figure 5a and 5b. We compute mean and confidence intervals for utilizing different numbers of decoders trained under single- or multiple-layer paradigm. The higher the SSIM, the better the decoder is able to retain content image structure. We can see that for both inference structures, as the decoder has an increasing number of layers, SSIM slightly goes down. This is expected because as the number



(a) SSIM of Stacked Decoders (b) SSIM of Accumulative Decoders (c) SSIM of Single Decoder

Figure 5. SSIM for various model structures and layers.

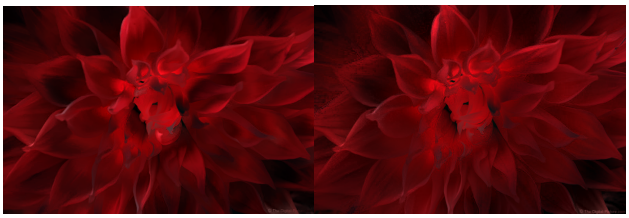
of decoders increases, the added deeper decoder will focus on transferring more color details from style image, and could cause a change in edge detection, which results in lower SSIM score. Also, for each number of decoders, stacked decoders achieve only slightly better SSIM then accumulative decoders. While SSIM indicates there is little difference in the two decoders' ability to retain the original content under different training paradigms, figure 3 clearly shows that the stack decoders transfer the content in a more aesthetically pleasing manner.

Comparison between Different Layers

To further probe the effect of each layer, we extract stylization result after passing in the content through each layer. From Figure 5c we see that for single layer decoders, decoder 4 has the highest SSIM value while decoder 1 has the lowest. This is consistent with what is shown in figure Figure 3c to Figure 3f. Figure 3c shows that decoder 1 carries more low-level information (e.g. colors) and as we move deeper into the layers, 3f shows that decoder 4 captures more semantic details, allowing it to better preserve the original structural information.

5.2. Different Smoothing Techniques

While the original PhotoWCT paper mainly focuses on affinity-based smoothing, it also mentions guided image filtering as a fast yet slightly degraded smoothing choice. We conducted some mini-experiments and found out that while smoothing with matting affinity can be better at preserving fine structures and subtle details in images, its performance is dependent on the balance factor λ, which controls the balance of the smoothness term and the fitting term. For the same lambda, it performs well on some photos (e.g. Figure 6) but overly smoothens some details in others (e.g. Figure 7). If more time allowed, we will construct automated tuning to further investigate how different hyperparameters affect smoothing quality.



(a) Affinity-based Smoothing (b) GIF Smoothing

Figure 6. Example Photo Matting Affinity Outperforms

(a) Affinity-based Smoothing (b) GIF Smoothing

Figure 7. Example Photo Guided Image Filter Outperforms

5.3. Conclusions

In conclusion, our study explores how different training and inference paradigm affect how well the decoders preserve the structural detail of the original photo. We discover that, with respect to content preservation, there is not much difference between single-layer training and multiple-layer training. With respect to color transferring, multiple-layer training still achieves more aesthetically pleasing results. We also conclude that as we move deep into the layers, the decoders gradually learns to convey more subtle and detailed information.

6. Challenges

Despite the promising results, our study faced several challenges:

6.1. Model architecture

Exploring different architecture designs (e.g., max-pooling, stride, etc.) for PhotoWCT can be complex and time-consuming. Identifying the most effective architecture for preserving information and achieving optimal results remains a challenge.

6.2. Training Challenge

The training of four different decoders may necessitate individualized attention, as each decoder might demand distinct learning rates for convergence. This situation introduces an added dimension of complexity to the tuning process and consequently increases the computational burden. Researchers should mitigate their impact on compute resource allocation.

7. Future Scope

7.1. New Metric Exploration

While SSIM serves as an effective metric for evaluating the preservation of details in transformed images, incorporating additional metrics could lead to a more comprehensive assessment of the model's performance.

Determining an appropriate metric for photorealistic style transfer remains challenging, as it is largely subjective. By addressing these challenges and identifying suitable metrics, we can further refine the PhotoWCT model, ultimately enhancing its performance in photorealistic image stylization tasks.

7.2. New Architecture Exploration

There are several innovations in this field lately, including but not limited to WCT^2 (Yoo et al., 2019) and PhotoWCT^2 (Chiu & Gurari, 2022).

The former method utilizes wavelet-based skip connections to perform fine-to-coarse feature transformations within a single autoencoder, while the latter approach incorporates high-frequency residuals. Notably, both of these techniques leverage a single decoder to achieve results comparable to those obtained with PhotoWCT.

It is worth considering new ideas that build upon these approaches to further improve their efficacy.

8. Acknowledgments

Our implementation is inspired from https://github.com/vidursatija/PhotoWCT and https://github.com/NVIDIA/FastPhotoStyle. We would also like to thank Jaejun Yoo, the author of WCT2, for providing extra training code (Yoo et al., 2019). Finally, we would like to thank our professor Jean Ponce and teaching assistant Ayush Jain for their great support and aspiring suggestions.

References

Bell, A. J. and Sejnowski, T. J. The "independent components" of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997. ISSN 0042-6989. doi: https://doi.org/10.1016/S0042-6989(97)00121-1. URL https://www.sciencedirect.com/science/article/pii/S0042698997001211.

Champandard, A. J. Semantic style transfer and turning two-bit doodles into fine artworks. *arXiv preprint arXiv:1603.01768*, 2016.

Chiu, T.-Y. and Gurari, D. Photowct2: Compact autoencoder for photorealistic style transfer resulting from blockwise training and skip connections of high-frequency residuals. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2868–2877, 2022.

Dosovitskiy, A. and Brox, T. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016.

Gatys, L., Ecker, A. S., and Bethge, M. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28, 2015.

Gatys, L. A., Ecker, A. S., and 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.

Heeger, D. J. and Bergen, J. R. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 229–238, 1995.

Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., and Song, M. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 694–711. Springer, 2016.

Levin, A., Lischinski, D., and Weiss, Y. A closed-form solution to natural image matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):228–242, 2007.

Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., and Yang, M.-H. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30, 2017.

Li, Y., Liu, M.-Y., Li, X., Yang, M.-H., and Kautz, J. A closed-form solution to photorealistic image stylization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 453–468, 2018.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Luan, F., Paris, S., Shechtman, E., and Bala, K. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4990–4998, 2017.

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

Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q., Pietikäinen, M., and Liu, L. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5117–5127, 2021.

Ulyanov, D., Lebedev, V., Vedaldi, A., and Lempitsky, V. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016.

Yoo, J., Uh, Y., Chun, S., Kang, B., and Ha, J.-W. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9036–9045, 2019.