Ultrafast Photorealistic Style Transfer via Neural Architecture Search

Jie An,[*†1] **Haoyi Xiong,**[†2] **Jun Huan,**[3] **Jiebo Luo**[1]
[1]University of Rochester, [2]Baidu Research, [3]StylingAI Inc.
{jan6, jluo}@cs.rochester.edu, xionghaoyi@baidu.com, lukehuan@shenshangtech.com

Abstract

The key challenge in photorealistic style transfer is that an algorithm should faithfully transfer the style of a reference photo to a content photo while the generated image should look like one captured by a camera. Although several photorealistic style transfer algorithms have been proposed, they need to rely on post- and/or pre-processing to make the generated images look photorealistic. If we disable the additional processing, these algorithms would fail to produce plausible photorealistic stylization in terms of detail preservation and photorealism. In this work, we propose an effective solution to these issues. Our method consists of a construction step (C-step) to build a photorealistic stylization network and a pruning step (P-step) for acceleration. In the C-step, we propose a dense auto-encoder named PhotoNet based on a carefully designed pre-analysis. PhotoNet integrates a feature aggregation module (BFA) and instance normalized skip links (INSL). To generate faithful stylization, we introduce multiple style transfer modules in the decoder and INSLs. PhotoNet significantly outperforms existing algorithms in terms of both efficiency and effectiveness. In the P-step, we adopt a neural architecture search method to accelerate PhotoNet. We propose an automatic network pruning framework in the manner of teacher-student learning for photorealistic stylization. The network architecture named PhotoNAS resulted from the search achieves significant acceleration over PhotoNet while keeping the stylization effects almost intact. We conduct extensive experiments on both image and video transfer. The results show that our method can produce favorable results while achieving 20-30 times acceleration in comparison with the existing state-of-the-art approaches. It is worth noting that the proposed algorithm accomplishes better performance without any pre- or post-processing.

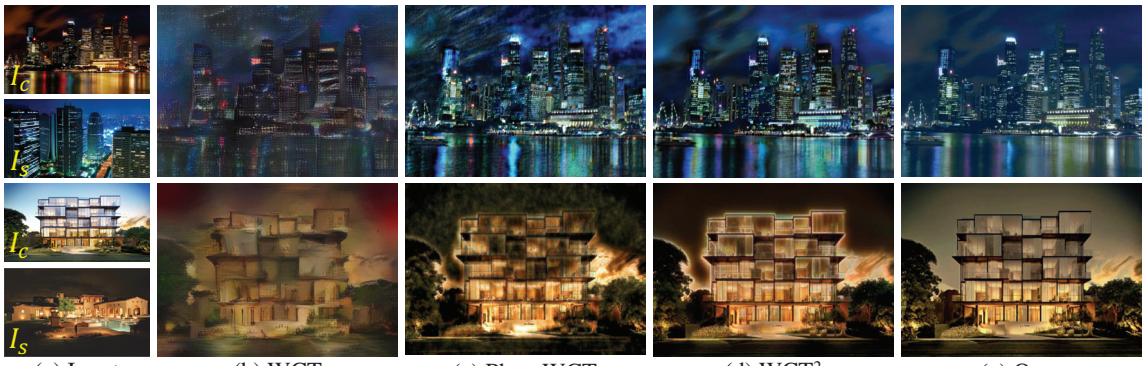
Introduction

Photorealistic style transfer is an image editing task aims at changing the style of a photo to a given reference. To be photorealistic, the produced image should preserve spatial details of the input and looks like a photo captured by a camera. For example, in Fig. 1, we transfer the night view photo from a warm color to cold while in the other example, a day-time photo is changed to a night-time one. In these examples, the scene of the input content keeps intact in the produced result. Unfortunately, artistic style transfer methods (Gatys, Ecker, and Bethge 2015; 2016; Johnson, Alahi, and Fei-Fei 2016; Ulyanov et al. 2016; Li et al. 2017; Huang and Belongie 2017; Sheng et al. 2018; Li et al. 2019) generally distort fine details (lines, shapes, borders) in images, which is necessary for producing art flavors in artistic scenarios but is not favored in photorealistic stylization. We illustrate the failure of artistic methods in photorealistic stylization cases with the example of WCT in Fig. 1 (b). More failure cases are available in supplementary materials.

Based on Gatys *et al.* (Gatys, Ecker, and Bethge 2016), Luan *et al.* (Luan et al. 2017) introduce a photorealistic loss term and adopts an optimization method to make style transfer. However, solving the optimization problem is time/computation consuming. To address this issue, Li *et al.* propose PhotoWCT (Li et al. 2018) which uses a feed-forward network to make style transfer. Although PhotoWCT applied multi-level stylization and uses unpooling operator as a replacement of upsampling to enhance the detail preservation of the network, the produced results still suffer from distortions as demonstrated in Fig. 1 (c). To overcome the remaining artifacts, they have to introduce close-formed post-processing to regulate the spatial affinity of the image. However, such post-processing is computationally expensive and causes the result over-smoothed. Recently, Yoo *et al.* (Yoo et al. 2019) proposed Wavelet Corrected Transfer (WCT^2) aims at eliminating post-processing steps while preserving fine details in transferred photos. Although using wavelets can increase the fidelity of signal recovery, WCT^2 need to rely on region masks of content and reference style photos to perform style transfer. If such region masks are disabled, as shown in Fig. 1 (d), the result of WCT^2 shows significant distortions. Since such region masks are hard to acquire for arbitrary photos (generally have to train specific networks to segment input photos and manually fine-tune the segmented results), the practical usage of WCT^2 is limited.

[*]This work was done when Jie An worked as an intern at Big Data Lab of Baidu Research.

[†]Equal contribution.



(a) Input (b) WCT (c) PhotoWCT (d) WCT^2 (e) Ours

Figure 1: Photorealistic style transfer results. Given (a) an input pair (I_c: content, I_s: style), we show results of (b) WCT (Li et al. 2017), (c) PhotoWCT (Li et al. 2018), (d) WCT^2 (Yoo et al. 2019), and (e) our method. Every result is produced *without* the assist of region masks and/or post-processing for a fair comparison. While the compared methods produce significant spatial distortions, the proposed approach achieves better style transfer results in terms of fine detail preservation and photorealism.

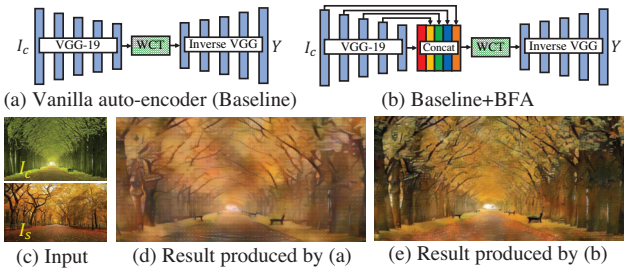
Regarding the network architecture, PhotoWCT and WCT^2 both adopt the same symmetric auto-encoder but use different downsampling and upsampling modules. However, general network architectures specifically designed for photorealistic style transfer have not been well investigated. This work fills this gap. Specifically, our algorithm consists of a network construction step (C-step) that introduces a highly-effective auto-encoder for photorealistic stylization, and a pruning step (P-step) is adopted in the following to compress the auto-encoder for acceleration. In C-step, we firstly conduct a carefully designed pre-analysis and introduce two architectural modules named *Bottleneck Feature Aggregation (BFA)* and *Instance Normalized Skip Link (INSL)* based on analyzed results. BFA, motivated by (Yu et al. 2018; Zhao et al. 2017), employs multi-resolution deep features to improve photorealistic stylization effects. INSL is the combination of the Skip Connection (SC) originated from U-Net (Ronneberger, Fischer, and Brox 2015) and the Instance Normalization (Ulyanov, Vedaldi, and Lempitsky 2016). INSL achieves high fidelity information recovery while avoiding "short circuit" phenomenons occurred when using SCs. Based on these modules, we constructed an asymmetric auto-encoder (named **PhotoNet**) with BFA and densely placed INSLs. Thanks to the proposed modules, our PhotoNet outperforms DPST (Luan et al. 2017), PhotoWCT and WCT^2 in terms of fine detail preservation. In P-step, we propose a Neural Architecture Search framework in a manner of teacher-student learning (namely StyleNAS). Here PhotoNet is the maximum architecture in our search space of NAS, where an evolution algorithm (Kim et al. 2017) is adopted to iterative prune removable operators (any operator except the VGG encoder and minimal basic operators to form a decoder) in PhotoNet. In each loop of the architecture search, we first mutate 20 new architectures. Each architecture contains a pre-trained VGG-19 (Simonyan and Zisserman 2014) as the encoder and the decoder is trained to reconstruct images. A validation process is adapted after training, where the performance of each architecture is evaluated by its similarity to the result of the oracle (*i.e.* , PhotoNet). To compress network architectures, we additionally introduce a network complexity loss to penalize time-consuming networks and finally get a bunch of highly-efficient and effective networks for photorealistic style transfer. We pick up one of them (named **PhotoNAS**) for comparison in this paper and more searched architectures and its results are available in supplementary materials.

Our contributions are two-fold. For photorealistic style transfer, PhotoNet/PhotoNAS are the first networks that *do not require any post-processing or region mask assistance*. PhotoNAS is surprisingly simple and highly-efficient with $356\times$ *speed up over PhotoWCT and* $24\times$ *over WCT*^{2} *on* 1024×512 *photos*. PhotoNAS quantitatively outperforms the state-of-the-art methods in terms of SSIM-Edge, SSIM-Whole, Gram Loss, and user preference percentage. Further experiments on video style transfer demonstrate its ability to stylize and produce stable videos without any specific modification. On the other hand, for Automatic Machine Learning (AutoML) and NAS, *our algorithm is the first that successfully adopts NAS to design style transfer networks for photorealistic rendering*, which expands the application area of NAS to the style transfer area.

Related Work

Style Transfer. Significant efforts have been made to image style transfer in the area of computer vision. Prior to the adoption of deep neural networks, several classical models based on stroke rendering (Hertzmann 1998), image analogy (Hertzmann et al. 2001; Shih et al. 2013; 2014; Frigo et al. 2016; Liao et al. 2017), or image filtering (Winnemöller, Olsen, and Gooch 2006) have been proposed to make a trade-off between quality, generalization, and efficiency for style transfer.

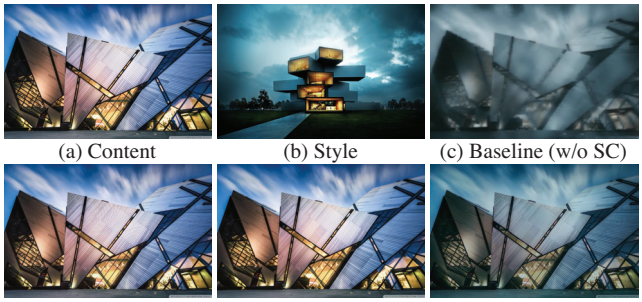
Gatys *et al.* (Gatys, Ecker, and Bethge 2015; 2016) first proposed to model the style transfer as an optimization problem minimizing deep features and their Gram matrices of neural networks, while these networks were designed to



(a) Vanilla auto-encoder (Baseline) (b) Baseline+BFA

(c) Input (d) Result produced by (a) (e) Result produced by (b)

Figure 2: Comparison between auto-encoders *with* and *without* BFA. (a) is the vanilla auto-encoder with WCT as the transfer module placed in the bottleneck, which is used as the baseline. (b) is the auto-encoder equipped with BFA module. (c) is the input content (I_c) and style (I_s) images. (d) and (e) are results produced by (a) and (b) respectively. Trees in (e) contain comparably more detailed branches and leaves.



(a) Content (b) Style (c) Baseline (w/o SC)

(d) Baseline+SC (e) WCT^2 (w/ HFCS) (f) Ours (w/ INSL)

Figure 3: Comparison of SC, HFCS and INSL. SC (d) causes the "short circuit" issue that removes stylization effects of the baseline (c). Similar failure case also exists in WCT^2 with HFCS turned on (e). The proposed INSL (f) can overcome the side effect of SC while enjoying enhanced detail preservation.

work well with artistic styles only. In photo style transfer scenarios, neural network approaches (Luan et al. 2017; Li et al. 2018) have been proposed to enable style transfer for photorealistic styles. These methods either introduce smoothness-based loss term (Luan et al. 2017) or utilize post-processing to smooth the transferred images (Li et al. 2018), which inevitably decreased fine details of images and increased time-consumption significantly. Recently, Yoo *et al.* (Yoo et al. 2019) proposed WCT^2, which allows transferring photorealistic styles without inefficient post-processing. However, WCT^2 has to work with the assist of region masks, which are hard to acquire and thereby limit its practical applications.

Image-to-image Translation. In addition to style transfer, photorealistic stylization has also been studied in image-to-image translation (Isola et al. 2017; Wang et al. 2018; Liu and Tuzel 2016; Taigman, Polyak, and Wolf 2017; Shrivastava et al. 2017; Liu, Breuel, and Kautz 2017; Zhu et al. 2017; Huang et al. 2018). The major difference between photorealistic style transfer and image-to-image translation is that photorealistic style transfer does not require paired training data (i.e., pre-transfer and post-transfer images). Of course, image-to-image translation can solve even more complicated task such as the man-to-woman and cat-to-dog adaption problems.

Discussion. The work most relevant to our study includes WCT, PhotoWCT, and WCT^2. WCT has been used for artistic stylization and the last two ones are for photorealistic stylization. Compared with PhotoWCT, the proposed method can avoid time-consuming post-processing and multi-round stylization while ensuring the effectiveness of style transfer. The main difference between our approach and WCT^2 is that the proposed algorithm allows transferring photo styles without any assist of region masks acquired by segmenting content and style inputs. Compared with PhotoWCT and WCT^2, the results produced by our method has considerably higher sharpness, fewer distortions and a remarkable reduction of computational cost.

Pre-analysis

To design effective modules/networks for photorealistic style transfer, we start with conducting a pre-analysis on architectural factors may influence stylization effect to propose useful network modules for the enhancement of stylization performance. We adopt a vanilla symmetrical auto-encoder as the baseline. For each studied module, we will compare its transfer results with the baseline in terms of visual effects and photorealism. More analyzed results are available in supplementary materials.

Feature Aggregation. Feature aggregation is a network module that concatenates multi-scale features produced by different layers of deep networks. Feature aggregation enables networks to integrate information from different field-of-views, thus may enhance low-level detail preservation of stylization that happens in high-level features. Based on this, we introduce a bottleneck feature aggregation (BFA) module to the auto-encoder. In detail, we first resize features from ReLU_1_1 to ReLU_4_1 to the size of ReLU_5_1 in the VGG encoder, then we concatenate them together at the bottleneck. Please refer to Fig. 2 (b) for details. We show the style transfer results produced by networks *with* and *without* BFA in Fig. 2 (d) and (e) respectively, which show that BFA can preserve more fine details (*e.g.* , more detailed tree branches and leaves in Fig. 2). To the best of our knowledge, we are the first that adopt the feature aggregation module to style transfer tasks.

Skip Link. The Skip Connection (SC) is first introduced by FCN (Long, Shelhamer, and Darrell 2015) and U-Net (Ronneberger, Fischer, and Brox 2015), where SC can significantly enhance their segmentation results. However, the auto-encoder equipped with SC generally lost its ability to produce stylized images since SC can make the transfer module at the bottleneck of the auto-encoder invalid. We call this issue *"short circuit"* phenomenon. As demonstrated in Fig. 3 (d), the image produced by the auto-encoder with SC totally lost stylization effects compared with that without SC (show in Fig. 3 (c)). The reason behind this

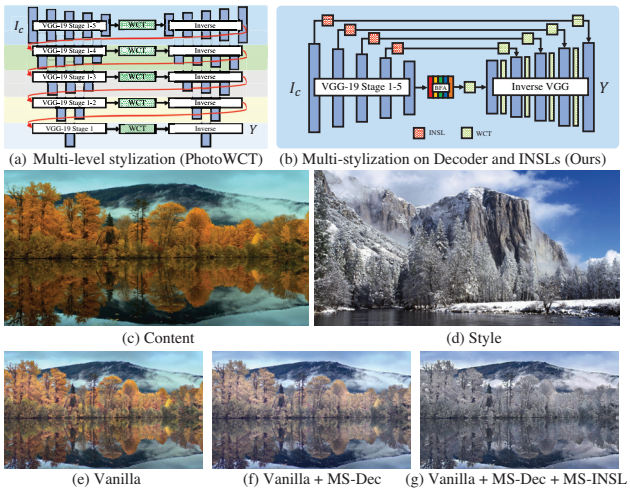


Figure 4: Multi-stylization Comparison. (a) is the multi-level stylization strategy used by WCT/PhotoWCT, which adopts five distinct auto-encoders in cascade to make style transfer. (b) is the architecture of our method. Please note that (b) equals to the auto-encoder in the top blue box in terms of computation cost. From (e) to (g), we progressively apply style transfer modules (*i.e.* WCT) at the bottleneck, decoder, and INSLs, where MS-Dec and MS-INSL denote placing transfer module at decoder and INSLs respectively. As demonstrated in (e-g), MS-Dec and MS-INSL enhance style transfer effects without sacrificing fine details of the content. Please see colors of leaves in (e-g).

Figure 5: Comparison of "Concat" and "Sum".

is that SCs placed at low-level layers of an auto-encoder will short circuit and block the information stream flow into transfer modules work at the bottleneck. Interestingly, as shown in Fig. 3 (e), we find that WCT^2 also fails to make stylization if turn their proposed High-Frequency Components Skip Links (HFCS) on and disable the input region masks. To solve this problem, we introduce the Instance Normalized Skip Links (namely INSL) as a replacement of the SC, which applies the Instance Normalization (Ulyanov, Vedaldi, and Lempitsky 2016) at skip connections. We find that INSL can alleviate the short circuit phenomenon and strengthen the detail preservation and distortion elimination abilities of photorealistic style transfer networks. Please refer to Fig 3 (f) for the result produced with INSLs.

Multi-stylization. Multi-stylization means make style transfer repeatedly. As shown in Fig. 4 (a), WCT and PhotoWCT adopt a strategy called *multi-level stylization*. They train five auto-encoders and make stylization for five rounds in a coarse-to-fine manner. Instead of that, WCT^2 proposes *progressive stylization*, which uses a single round auto-encoder but progressively executes style transfer modules multi times at every part of the auto-encoder. Following WCT^2, we adopt a single-round multi stylization strategy but only transfer features at the decoder and INSLs. Fig. 4 (b) illustrates our strategy. As demonstrated in Fig. 4 (e-g), MS-Dec and MS-INSL can significantly improve the produced results in terms of stylization effects. Moreover, applying style transfer modules at INSLs (Fig. 4 (g)) can further eliminate the short circuit phenomenon caused by SC and strengthen the stylization effects.

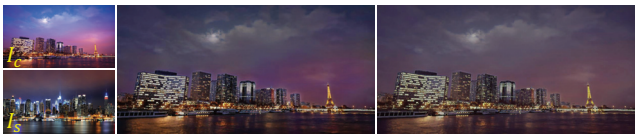


Figure 6: Comparison of "Upsampling" and "Unpooling".

Figure 7: Comparison of using AdaIN and WCT as transfer module. Using WCT as transfer module (c) achieves more faithful photorealistic stylization effects against using AdaIN (b).

Concat *v.s.* Sum. The choice of "concat" and "sum" operators when using skip links is a factor that may influence the performance of auto-encoders. However, we find that using "concat" generally has no specific difference against using "sum" except little style fluctuation. Please refer to Fig. 5 (b) (c) for comparison.

Upsampling *v.s.* Unpooling. PhotoWCT argues that the unpooling tends to make the network produce fewer distortions. However, we find that these two operators produce almost the same results in our settings. Please refer to Fig. 6 (b) (c) for comparison.

WCT *v.s.* AdaIN. WCT and AdaIN are two widely used transfer modules that come from artistic style transfer. As demonstrated in Fig. 7 (b) (c), WCT can produces more faithful transfer results. We think this is because AdaIN need to work with the auto-encoder trained in a more complicated way. However, we just train the decoder to reconstruct images to facilitate the following pruning step.

C-Step

Based on the analysis on architecture components that have significant influence on photorealistic style transfer effects, we construct an auto-encoder named *PhotoNet*.

The C-step part (*i.e.* , grey and blue boxes) in Fig. 8 shows

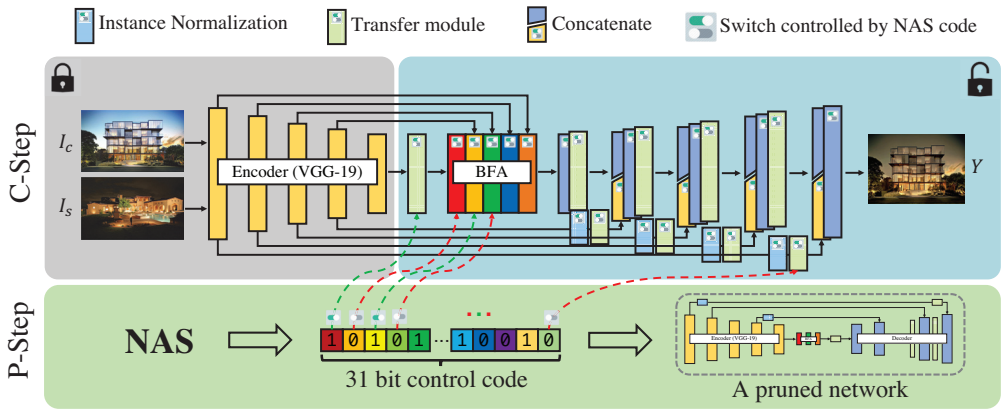


Figure 8: Framework of the proposed method. Our method consists of a C-step and P-step. In C-step, we construct a highly effective dense auto-encoder. In P-step, we propose a neural architecture search (StyleNAS) algorithm to automatically prune the auto-encoder. In each loop of the auto-pruning, the encoder part (in grep box) is fixed while 31 operators in the blue box is controlled by 0/1 code to turn off/on. Please note that yellow and cyan rectangles represent sequential convolution operators.

Table 1: Differences between our approach and other methods.

	DPST	PhotoWCT	WCT^2	Ours
Learning-free	×	✓	✓	✓
No post-processing	✓	×	✓	✓
No pre-mask	×	✓	×	✓
Efficient	×	×	✓	✓

the architecture of PhotoNet. The encoder of PhotoNet is a VGG-19 that pre-trained on ImageNet dataset. The decoder is trained to invert deep features of the encoder back to images. In the bottleneck of PhotoNet, as demonstrated in the pre-analysis part, we place a BFA module to make use of multi-scale features. Between the encoder and decoder, we introduce INSLs to transport information from encoder stages (ReLU_1_1 to ReLU_4_1 in VGG-19) to their corresponding decoder layers. our INSL has two advantages: on the one hand, INSL enhances the detail preservation ability of PhotoNet, hereby improves photorealism. On the other hand, the equipped instance normalization can surprisingly weaken short circuit issue caused by skip connections. To improve photorealistic style transfer performance, we densely apply transfer modules (*i.e.* , WCT) at the bottleneck, every stage of the decoder, and INSLs. Interestingly, making style transfer at INSLs further eliminated the short circuit phenomenon caused by skip links.

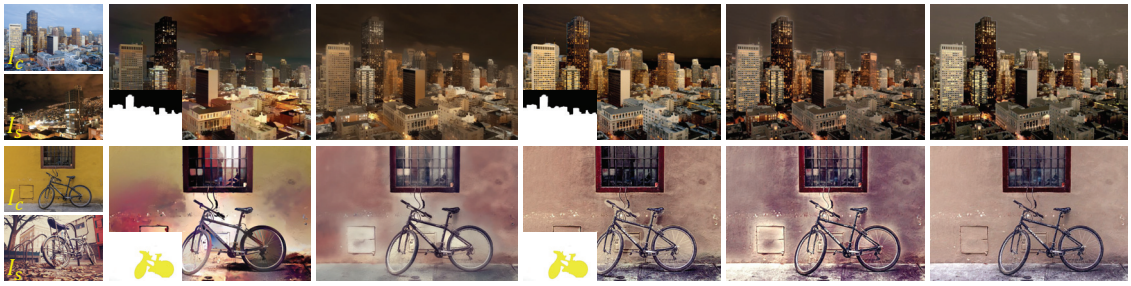
During training, all transfer modules are temporarily removed and the encoder is fixed. The decoder (without transfer modules) is trained on MS_COCO dataset (Lin et al. 2014) to invert deep features of the encoder back to images. With the trained network, our PhotoNet directly takes a content photo and a style photo as input and outputs a style transferred photo. It is worth mentioning that our PhotoNet and the pruned version that will be introduced in the next part do not need any pre-conditioned region masks as DPST and WCT^2 do. Thanks to the strong detail preservation ability, our method enjoys fewer distortions against state-of-the-art algorithms while avoiding the usage of any time-consuming post-processing. Based on above-mentioned advantages, PhotoNet allows end-to-end photorealistic style transfer.

Please refer to Fig. 4 (g) for results of fully equipped PhotoNet. More results are available in supplementary materials. It is worth mentioning that PhotoNet is 7 and 107 times faster than WCT^2 (without counting the time for making segmentation masks) and PhotoWCT respectively.

P-Step

To further accelerate PhotoNet, a P-step is proposed to automatically prune PhotoNet and discover more efficient style transfer networks for photorealistic rendering while maintaining stylization effects of the PhotoNet. We achieve this by using PhotoNet as the maximum architecture and introducing a neural architecture search method named StyleNAS in a manner of teacher-student learning for automatic pruning. Given the MS_COCO as the training dataset and a validation dataset with 40 content and style photos, we first train PhotoNet as the *Supervisory Oracle* for the subsequent architecture search. The P-step consists of the following three key components.

Search Space. We use fully equipped PhotoNet and all of its simplified versions (remove some operators) as the search space. Please refer to grey and blue boxes in Fig. 8 (*i.e.* C-step part) for details. In each loop of the neural architecture search, 31 options of operators have been remained to form a functional architecture, while one can open/close a bit to determine use/ban an operator. We encode any architecture in this space using a string of 31-bits. For example, the searched PhotoNAS architecture is encoded as "0101000000100000000000000001111" in our setting. In this way, StyleNAS can search new architectures in a combinatorial manner from totally $2^{31} \approx 2.1 \times 10^9$ possible architectures. We hereby denote the search space as Θ which



(a) Input (b) DPST (w/ Mask) (c) PhotoWCT (w/ smooth) (d) WCT^2 (w/ Mask) (e) Ours (PhotoNet) (f) Ours (PhotoNet-Auto)

Figure 9: Visual comparison to state-of-the-art methods. (a) is the input content (I_c) and style (I_s) photos. DPST (b) and WCT^2 (d) have to run with the assist of regional masks (show in left-bottom corner) and the result of PhotoWCT (c) are produced with post-processing. Our methods ((e) PhotoNet, (f) PhotoNAS) do not need any pre- and post-processing.

Table 2: Quantitative evaluation results for stylization methods. Higher SSIM-Edge and SSIM-Whole scores mean the measured image is more similar to the input content photo in terms of fine details. A lower Gram Loss denotes the evaluated image has more similar visual effects to the style photo. Here results of DPST and WCT^2 are produces without the assist of segmentation maps for a fair comparison.

Method	DPST	PhotoWCT	PhotoWCT+Smooth	WCT^2	Ours(PhotoNet)	Ours(PhotoNAS)
SSIM-Edge ↑	0.6395	0.5690	0.6391	0.6112	0.6922	**0.6932**
SSIM-Whole ↑	0.5139	0.5013	0.5005	0.4723	**0.7047**	0.6728
Gram Loss ↓	1.4143	1.2130	2.1660	**1.1244**	1.1270	1.7565

refers to the full set of all architectures.

Search Objectives. To obtain highly-efficient and effective architectures from Θ, we adopt three search objectives: (i) the loss of knowledge distillation from a pre-trained supervisory oracle (PhotoNet), (ii) the perceptual loss of the produced images and oracle, and (iii) the percentage of operators used in the architecture. The knowledge distillation loss reflects image reconstruction errors in a supervisory manner. We write the overall search objective as

$$\mathcal{L}(\theta) = \alpha \cdot \mathcal{E}(\theta) + \beta \cdot \mathcal{P}(\theta) + \gamma \cdot \mathcal{O}(\theta), \quad (1)$$

$$\mathcal{E}(\theta) = \underset{I \in \mathbb{V}}{\text{mean}} \|I_\theta - I_{oracle}\|_F, \quad (2)$$

$$\mathcal{P}(\theta) = \underset{I \in \mathbb{V}}{\text{mean}} \sum_{i=1}^{5} \|\Phi_i(I_\theta) - \Phi_i(I_{oracle})\|_F, \quad (3)$$

where $\theta \in \Theta$ refers to an architecture drawn from the space; $\mathcal{L}(\theta)$ stands for the overall loss of the architecture θ; $\mathcal{E}(\theta)$ refers to the reconstruction error between the style-transferred images produced by the network with the architecture θ and those produced by the supervisory oracle; $\mathcal{P}(\theta)$ estimates the *Perceptual Loss* using a trained network with the architecture θ and the oracle; $\Phi_i(\cdot)$ denotes the output of the i^{th} stage of the ImageNet pre-trained VGG-19; $\mathbb{V}$ denotes the validation set with 40 content and style photos; $\mathcal{O}(\theta)$ estimates the percentage of operators used in θ of 31-bins; α, β and γ are a pair of hyper-parameters to make trade-off between these three factors.

Search Strategies. Our search strategies are derived from (Kim et al. 2017), where parallel evolutionary strategies with a map-reduce alike update mechanism have been used to iteratively improve the searched architectures from random initialization. From the search space Θ, the StyleNAS algorithm first randomly draws P architectures $\{\theta_1^1, \theta_1^2, \theta_1^3 \ldots \theta_1^P\} \subset \Theta$ (represented as P 31-bit strings) for the 1^{st} round of iteration, where P refers to the number of populations desired. On top of the parallel computing environment, the algorithm maps every drawn architecture to one specific GPU card/worker, then trains the style transfer networks for image reconstruction (with WCT modules temporarily turned off), and evaluates the performance of trained networks (using the objectives in Eq 3). With the search objective estimated, every worker updates a shared *population set* using the evaluated architecture in an asynchronous manner, and generates a new architecture through *mutating* the best one in a subset of architectures drawn from the *population set*. With the newly generated architecture, the worker starts a new iteration of training and evaluating for the update and discards the oldest model from the *population set*. During the whole process, the algorithm keeps maintaining a *history set* of architectures that have been explored with their objectives estimated, all in an asynchronous manner. After T rounds of iterations on every worker, the algorithm returns the architecture with the minimal objectives from the overall *history set* by the end of the algorithm. Please refer to the supplementary for more details.

Experimental Results

In this section, we show the result comparison of our algorithm with state-of-the-art photorealistic stylization methods, i.e., DPST, PhotoWCT, and WCT^2 in terms of visual

effects and time consumption. More comparison results, detailed experimental settings, user study results, video transfer results, and our failure cases are available in supplementary materials. All the source code will be made released in the future.

Visual Comparison We testify the effectiveness of the proposed method by the comparison with the photorealistic stylization results of DPST, PhotoWCT, and WCT^2. Since DPST and WCT^2 have to run with pre-acquired regional masks, we make comparison on images and corresponding segmentation masks provided by DPST in this part. Additionally, we add two post-processing to PhotoWCT as suggested by its paper. Note that results of our approaches (PhotoNet and PhotoNAS) do not involve any pre- and post-processing.

As shown in Fig. 9, results of DPST contains significant artifacts and are comparably over-smoothed. For example, textures of buildings in the upper photo and details of bicycle wheels in the bottom image are blurred. Moreover, wall and ground in the bottom image show undesirable colors. Although results of PhotoWCT (w/ smooth) (Fig. 9 (c)) have alleviated artifacts, they still suffer from distortions and create blurry images since they have to use smooth-oriented post-processing to decrease those artifacts. WCT^2 make some advances upon previous two methods in terms of detail preservation by applying regional masks. However, WCT^2 introduces a new drawback that the produced images usually have visual style mismatch at the boundary of different regions. Even worse, if those masks are not accurate enough, WCT^2 tends to generate images with considerable artifacts which significantly hurt the photorealism of produced images. Please *zoom-in* in Fig. 9 (d) to see skylines in the upper example and bicycle outlines painted on the wall in the bottom result. Foregrounds of the result by WCT^2 look like are pasted on the background, which is non-photorealistic. Fig. 9 (e) and (f) show results of our methods. PhotoNet achieves effective photorealistic stylization and faithful detail preservation. The results of PhotoNAS maintains the stylization effects of PhotoNet and in the meantime, further eliminates remained distortions. It is worth mentioning that PhotoNAS achieves such a result with only 1/5 time-consumption. Note that results of PhotoNet and PhotoNAS are produced without any pre- and post-processing while other methods use pre- (DPST, WCT^2) or post-processing (PhotoWCT). Please refer to Fig. 1 for comparison without pre-/post-processing, which additionally verified the effectiveness of our method.

Quantitative Comparison. Inspired by WCT^2, we adopt structural similarity (SSIM) index between the original content photo and the produced result to measure the detail preservation ability (*i.e.* photorealism) of methods. We compute SSIM on whole images (named SSIM-Whole) and their holistically-nested edge responses (Xie and Tu 2015) (named SSIM-Edge). To evaluate photorealistic stylization effects, we compute the Gram matrix difference (VGG style loss) following WCT.

Given a validation dataset contains 73 content and style photo pairs, we quantitatively evaluate the performance of the proposed and state-of-the-art methods by computing the above-mentioned metrics on this validation set. We show the quantitative comparison result in Tab. 2. The proposed PhotoNet and PhotoNAS achieve better scores in terms of SSIM-Whole and SSIM-Edge respectively, which means our methods have remarkably improved detail preservation ability. Tab 2 shows that the Gram Loss of our PhotoNet is a little higher than WCT^2. We argue this is due to the improvement of detail preservation would inevitably raise the Gram Loss. Such an assertion is also verified by the fact that the Gram Loss of PhotoWCT largely increased when applying smooth post-processing.

Table 3: Computing-time comparison.

Method	DPST	PhotoWCT	WCT^2	PhotoNet	PhotoNAS
256×128	114.11	4.07	4.42	0.76	**0.13**
512×256	293.28	20.72	5.28	0.86	**0.16**
768×384	628.24	53.05	6.30	0.95	**0.22**
1024×512	947.61	133.90	7.69	1.06	**0.32**

Computational Time Comparison. We conduct a computing time comparison against the state-of-the-art methods to demonstrate the efficiency of the proposed and searched network architectures. All approaches are tested on the same computing platform which includes an NVIDIA P100 GPU card with 16GB RAM. The time consumption of DPST, PhotoWCT, and WCT^2 are evaluated by running officially released code with their default settings. We compare the computing time on content and style images with different resolutions. As Table 1 shows, PhotoNet achieves $6\times$ faster against WCT^2 and PhotoNAS are almost 20-30$\times$ faster than WCT^2. Surprisingly, after the P-step, only 7 operators are left among searched ones.

Conclusion

In this paper, we present a two-stage method to address the photorealistic style transfer problem. In the first step, we analyze the influence of commonly used network architectural components on photorealistic style transfer. Based on that, we construct PhotoNet, which utilizes instance normalized skip links (INSL), bottleneck feature aggregation (BFA), and multi-stylization on decoder and INSLs, to generate rich-detailed and well-stylized images. In the P-step, we introduce a network pruning framework for photorealistic stylization adopting a neural architecture search (StyleNAS) method and teacher-student learning strategy. With the novel pruning method, we discover PhotoNAS, which is surprisingly simple and keeps the stylization effects almost intact. Our extensive experiments in terms of visual, quantitative, and computing time comparison show that the proposed approach has a strengthened ability to remarkably improve the stylization effects and photorealism while reducing the time consumption dramatically. Our study also expands the application area of NAS to photorealistic style transfer. In our future work, we plan to 1) design novel NAS method specifically for style transfer task and 2) extend the work to other generative models such as generative adversarial networks and other low-level vision tasks.

References

Frigo, O.; Sabater, N.; Delon, J.; and Hellier, P. 2016. Split and match: example-based adaptive patch sampling for unsupervised style transfer. In *CVPR*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *CVPR*.

Hertzmann, A.; Jacobs, C. E.; Oliver, N.; Curless, B.; and Salesin, D. H. 2001. Image analogies. In *SIGGRAPH*.

Hertzmann, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH*.

Huang, X., and Belongie, S. J. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*.

Huang, X.; Liu, M.-Y.; Belongie, S.; and Kautz, J. 2018. Multimodal unsupervised image-to-image translation. In *ECCV*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*.

Kim, Y.-H.; Reddy, B.; Yun, S.; and Seo, C. 2017. Nemo: Neuro-evolution with multiobjective optimization of deep neural network for speed and accuracy. In *ICML 2017 AutoML Workshop*.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017. Universal style transfer via feature transforms. In *NIPS*.

Li, Y.; Liu, M.-Y.; Li, X.; Yang, M.-H.; and Kautz, J. 2018. A closed-form solution to photorealistic image stylization. In *ECCV*.

Li, X.; Liu, S.; Kautz, J.; and Yang, M.-H. 2019. Learning linear transformations for fast arbitrary style transfer. In *CVPR*.

Liao, J.; Yao, Y.; Yuan, L.; Hua, G.; and Kang, S. B. 2017. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: common objects in context. In *ECCV*.

Liu, M.-Y., and Tuzel, O. 2016. Coupled generative adversarial networks. In *NIPS*.

Liu, M.-Y.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. In *NIPS*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Luan, F.; Paris, S.; Shechtman, E.; and Bala, K. 2017. Deep photo style transfer. In *CVPR*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*.

Sheng, L.; Lin, Z.; Shao, J.; and Wang, X. 2018. Avatar-net: multi-scale zero-shot style transfer by feature decoration. In *CVPR*.

Shih, Y.; Paris, S.; Durand, F.; and Freeman, W. T. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics* 32(6):200.

Shih, Y.; Paris, S.; Barnes, C.; Freeman, W. T.; and Durand, F. 2014. Style transfer for headshot portraits. *ACM Transactions on Graphics* 33(4):148.

Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; and Webb, R. 2017. Learning from simulated and unsupervised images through adversarial training. In *CVPR*.

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

Taigman, Y.; Polyak, A.; and Wolf, L. 2017. Unsupervised cross-domain image generation. In *ICLR*.

Ulyanov, D.; Lebedev, V.; Vedaldi, A.; and Lempitsky, V. S. 2016. Texture networks: feed-forward synthesis of textures and stylized images. In *ICML*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance normalization: the missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*.

Winnemöller, H.; Olsen, S. C.; and Gooch, B. 2006. Real-time video abstraction. *ACM Transactions on Graphics* 25(3):1221–1226.

Xie, S., and Tu, Z. 2015. Holistically-nested edge detection. In *ICCV*.

Yoo, J.; Uh, Y.; Chun, S.; Kang, B.; and Ha, J.-W. 2019. Photorealistic style transfer via wavelet transforms. In *ICCV*.

Yu, F.; Wang, D.; Shelhamer, E.; and Darrell, T. 2018. Deep layer aggregation. In *CVPR*.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *CVPR*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.