
StyleNAS: An Empirical Study of Neural Architecture Search to Uncover Surprisingly Fast End-to-End Universal Style Transfer Networks

Jie An*

School of Mathematical Sciences
Peking University
jie.an@pku.edu.cn

Haoyi Xiong*

Big Data Lab
Baidu Research
xionghaoyi@baidu.com

Jinwen Ma

School of Mathematical Sciences
Peking University
jwma@math.pku.edu.cn

Jiebo Luo

Department of Computer Science
University of Rochester
jluo@cs.rochester.edu

Jun Huan[†]

Big Data Lab
Baidu Research
huan.jun@baidu.com

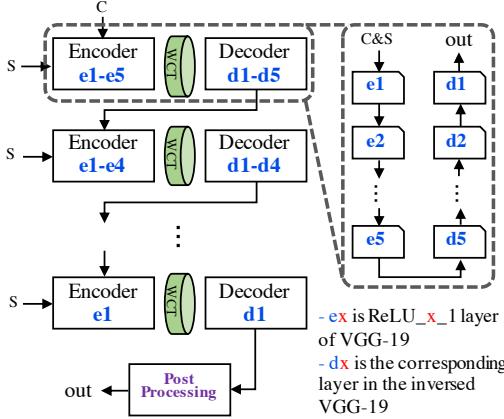
Abstract

Neural Architecture Search (NAS) has been widely studied for designing discriminative deep learning models such as image classification, object detection, and semantic segmentation. As a large number of prior work have been obtained through the manual design of architectures in the fields, NAS is usually considered as a supplement approach. In this paper, we have significantly expanded the application areas of NAS by performing an empirical study of NAS to search generative models, or specifically, auto-encoder based universal style transfer, which lacks systematic exploration, if any, from the architecture design aspect. In our work, we first designed a search space where common operators for image style transfer such as VGG-based encoders, whitening and coloring transforms (WCT), convolution kernels, instance normalization operators, and skip connections were searched in a combinatorial approach. With a simple yet effective parallel evolutionary NAS algorithm with multiple objectives, we derived the first-of-its-kind *end-to-end* deep networks for universal photorealistic style transfer. Comparing to Random Search, a NAS method that is gaining popularity recently, we demonstrated that carefully designed search strategy leads to much better architecture design.

Finally compared to existing universal style transfer networks for photorealistic rendering such as PhotoWCT that stacks multiple well-trained auto-encoders and WCT transforms in a *non-end-to-end* manner, the architectures designed by StyleNAS produce better style-transferred images with details preserving, using a tiny number of operators/parameters, and enjoying two orders of magnitude inference time speed-up.

*Equal contribution

[†]Corresponding author



(a) PhotoWCT: Stacked AEs with Transform Modules.

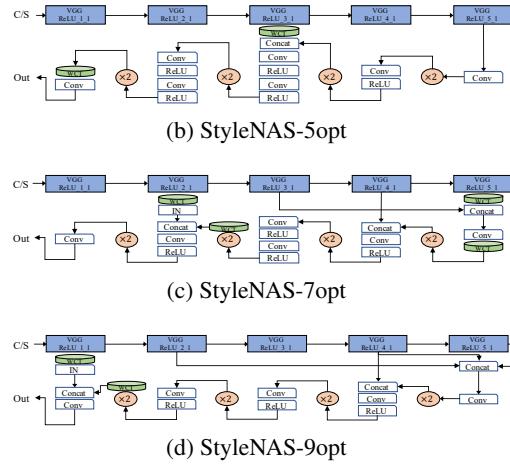


Figure 1: Comparison of architectures. The PhotoWCT network (19) adopts an Auto Encoder (AE) architecture. PhotoWCT uses VGG-19 (34) as the encoder and uses the reverse of VGG as the structurally symmetric decoder. PhotoWCT then stacks multiple instances of AE with the WCT transform modules (18) and a post-processing procedure for style transfer (shown in (a)). In our work we use a pre-trained VGG-19 (34) as the encoder and use NAS to optimize the decoder architecture. We identified three end-to-end architectures, namely StyleNAS-5opt, StyleNAS-7opt, and StyleNAS-9opt, using 5, 7, and 9 operators in the decoder. The identified StyleNAS architectures are shown in (b), (c), and (d) respectively.

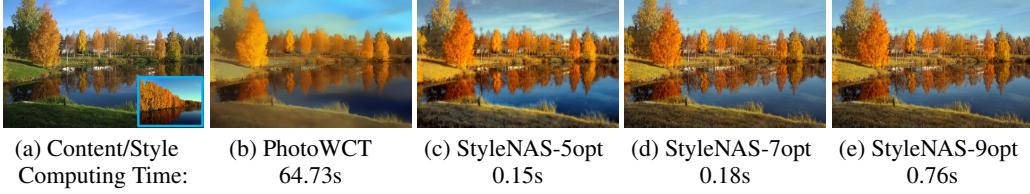


Figure 2: Visual comparison between PhotoWCT (19) and three searched architectures. The StyleNAS networks generate better results against PhotoWCT (19) in terms of detail preservation and stylization visual effects. (zoom-in for details of grassland and trees). Note that the computing time is obtained by evaluating on 768×384 images.

1 Introduction

Neural network architecture design lay at the center of deep neural network development (17). AlexNet (16), VGG (34), ResNet (9), DenseNet (13) are classical examples that facilitated the fast adaptation of deep neural networks in solving industry-scale real-world problems.

For a wide spectrum of discriminative learning tasks, such as image classification, object detection, and etc., Neural Architecture Search (NAS) (38; 39; 24; 22) is a very successful technique to design customized neural network architectures for a given dataset. As a large amount of prior work has been obtained through the manual design of architectures in the fields, NAS is usually considered as a supplement approach where architectures designed by NAS resemble high-level similarity to existing ones but with improved metrics such as top-1 accuracy. In this paper, we have significantly expanded the application areas of NAS by performing an empirical study of NAS to search generative models, or specifically, auto-encoder based universal style transfer for photorealistic rendering (25; 19; 28), which lacks systematic exploration, if any, from the architectural design aspect.

Universal style transfer is an image editing task that uses the visual styles of reference images to render arbitrary input content images. The state-of-the-art neural network for photorealistic stylization is PhotoWCT (19), which adopts a *non-end-to-end architecture* as shown in Figure 1(a). In terms of architecture, PhotoWCT stacks multiple locally trained auto-encoders (AEs) in a row while keeping style transfer modules (e.g., whitening and coloring transforms, WCT (18)) inserted intermediately. PhotoWCT produces style-transferred images but with missing details (please see also in the example

shown in Figure 2(a) and (b)). Computing cost with PhotoWCT is extremely high due to the stack of duplicated AEs in a non-end-to-end learning fashion and a slow post-processing step.

The success of PhotoWCT (19) demonstrated the effectiveness of some basic operators, such as VGG-based encoders, WCT modules, and convolution kernels for style transfer tasks. The high computational cost and the lost of details in generated images motivated our current work of using NAS to build-up novel and better architectures through reconnecting basic operators.

Our Work. In this work, we performed an empirical study of NAS to search for end-to-end deep neural networks for photorealistic universal style transfer with improved quality and computational efficiency. We first constructed a search space with 31 operators. There is hence a total of $2^{31} \approx 2.1 \times 10^9$ possible architectures, using common operators of neural networks from existing work on universal style transfer. We then designed three key objective functions for NAS, including (i) the reconstruction errors of style-transferred images compared to a pre-trained oracle such as PhotoWCT or handcrafted style transfer networks, (ii) the perceptual loss of the produced images and oracle, and (iii) the percentage of operators used in the architecture. With a simple yet effective parallel evolutionary NAS algorithm (15) that optimizes the three objectives, we derived a group of highly efficient and effective end-to-end architectures from the search space. We made the following technical contributions:

- We designed the StyleNAS algorithm and have found *the first-of-this-kind end-to-end architectures* for universal style transfer. Compared to PhotoWCT (18) that employs a *non-end-to-end* neural network architecture with time-consuming components, StyleNAS provides better style-transferred images with detail preserving and end-to-end training. To our surprise StyleNAS identified a much simpler decode with as small as 5 operators and practically had a $80\times \sim 300\times$ inference time speed-up. With a latency of about 0.05s (on 256×128 images) we are close to offer real-time photorealistic style-transfer images rendering on mobile settings.
- With simple evolutionary search strategies, in our experimental study we observed that StyleNAS can converge in both *search objectives* and *architectures*. We believe this is the first reported case of architecture convergence for NAS search with evolutionary computing. We have compared StyleNAS with the Random Search strategy that is gaining popularity recently (36) and found that StyleNAS delivers much better results. StyleNAS reduces the number of operators needed to form effective architectures that provide better image quality for photorealistic style transfer tasks.

In addition to the above contributions, our work successfully expended the application domain of NAS from automated design for discriminative deep models, to architecture search for generative models which has few investigations. To the best of our knowledge, it is the first work to study NAS for universal style transfer networks.

2 Related Work

We first review the most relevant work to our study and discuss the contribution made by our work.

2.1 Neural Network Architecture Search

NAS became a mainstream research topic since Zoph and Le (38) identified state-of-the-art recurrent cells on Penn Treebank and highly competitive architectures on CIFAR-10 using Reinforcement Learning (RL). Various RL methods have been successfully applied to NAS including vanilla policy gradient (2; 3), Proximal Policy Optimization (PPO) (39; 27) and Q-learning (1; 37). An alternative approach is to use evolution algorithm (30; 29; 23), maintaining and evolving a large population of neural architectures. In contrast to aforementioned gradient-free optimization methods, Liu et al. (24) proposed a gradient-bases search strategy based on continuous relaxation of architecture representation. Other gradient-based approaches include Neural Architecture Optimization (NAO) (26) and ProxylessNAS (4).

The research line of NAS search spaces has been largely influenced by the progress of the new architectures that are manually designed by experts. For example, Zoph et al. (39) and Zhong (37) proposed a search space based on the normal and reduction cells invented in Google Inception model, and further influenced the search space of many later works (22; 27; 29; 24; 26).

2.2 Deep Neural Networks for Style Transfer

Although significant efforts have been made to image style transfer in the area of computer vision, there is a very limited study on universal style transfer, especially for architecture design. Below we

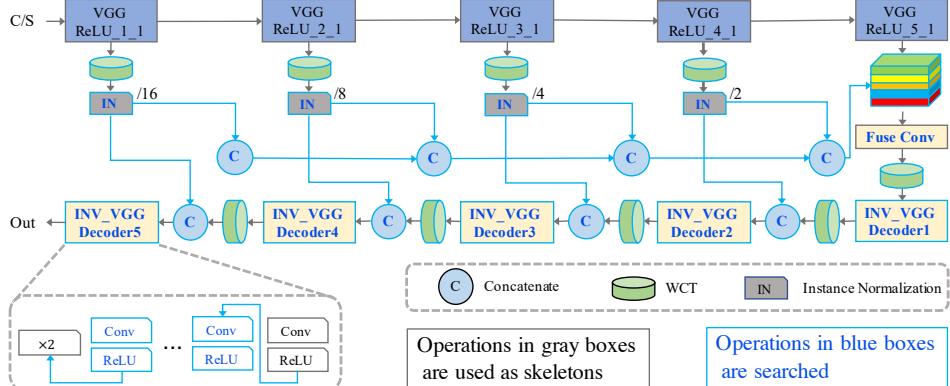


Figure 3: **Search space of the proposed StyleNAS.** We build a simple network skeleton with up-sampling operators coupled with a convolution layer and let NAS to determine whether to use other operators and connections.

briefly overview recent research progresses for style transfer. Prior to the adoption of deep neural networks, several classical models based on stroke rendering (10), image analogy (11; 33; 32; 6; 20), or image filtering (35) have been proposed to make a trade-off between quality, generalization, and efficiency for style transfer.

Gatys *et al.* (7; 8) 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 work well with artistic styles only. In addition to artistic stylization, neural network approaches (25; 19) have been proposed to enable style transfer for photorealistic styles. These methods either introduce smoothness-based loss term (25) or utilize post-processing to smooth the transferred images (19), which inevitably decreases the sharpness of images and increases the time-consumption significantly.

2.3 Discussion

The works that is most relevant to our study includes style transfer networks (18; 14; 19) and the neural architecture search (NAS) algorithms (38; 26; 15). Our work also falls in the category of leveraging NAS algorithms to search architectures for style transfer from an inspired search space, but differs in the following ways:

We carefully design the first search space for universal style transfer networks with accelerated search strategies. While existing approaches have to pass multiple rounds of pre-trained auto-encoders with intermediate transform modules, the new search space *de facto* forms networks using one pass of auto-encoder, where style transfer modules can be placed at any position of the decoder. Furthermore, though the search strategies of NAS were originally derived from (15), we provided a map-reduce friendly update mechanism to accelerate the parallel search.

We contributed to both NAS and universal style transfer. We extent the applications domain of NAS from discriminative learning to generative models, while simplifying the architectures used for photorealistic style transfer with better performance than the networks designed by human experts (19). Empirical studies have been done to evaluate NAS and validate the searched architectures.

3 Methodology

Given the MS_COCO (21) as the training dataset and a validation dataset with 40 content and style image pairs, we first train a handcrafted style transfer network as the *Supervisory Oracle* for the subsequent architecture search. Our proposed StyleNAS algorithm consists of the following three key components.

Search Space. We design a *search space* (shown in Figure 3) to parametrize the auto-encoder based universal style transfer networks for photorealistic rendering. Inspired by (18; 19), a fixed VGG-19 (34) pre-trained on Image-Net (5) is used as the encoder, while multiple convolution layers coupled with up-sampling operators have been stacked as the decoder. In addition to convolution layers followed by ReLU activation functions, WCT modules (18; 19), skip connections, instance normalization modules, and concatenate functions are all used to connect these convolution layers and/or build the style transfer architectures. As was shown in Figure 3, 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 StyleNAS-7opt architecture is encoded as “01010000001000000000000001111” in this 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 refers to the full set of all architectures.

Search Objectives. To obtain highly efficient and effective architectures from the search space, we design three new search objectives: (i) the loss of knowledge distillation from a pre-trained supervisory oracle, (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

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

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

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

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 Image-Net (5) pre-trained VGG-19 (34); \mathbb{V} denotes the validation set with 40 image pairs; $\mathcal{O}(\theta)$ 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 (15), 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 proposed NAS algorithm first randomly draws P architectures $\{\theta_1^1, \theta_1^2, \theta_1^3 \dots \theta_1^P\} \subset \Theta$ (represented as P 31-bit strings) for the 1st 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.

Note that, rather than proposing specific NAS algorithms for style transfer tasks through the combination of existing approaches, this work aims at investigating the feasibility of using NAS to design new architectures for style transfer tasks and validating the performance of searched architectures, so as to pursue new understandings and insights to NAS for generative models and universal style transfer networks for photorealistic rendering.

4 Experiments and Results

In this section, we introduce the experiment settings of StyleNAS and show the result comparison of StyleNAS networks with state-of-the-art photorealistic style transfer methods, i.e., PhotoWCT (19) and Random Search in terms of visual effects and computation time. Our results show that StyleNAS can produce style-transferred images with better image quality than PhotoWCT (19), while the searched architectures only consume 0.15%~1% computation time (run with StyleNAS-5opt) of PhotoWCT (19)³. We demonstrate that StyleNAS networks are better than Random Search in photorealistic style transfer cases.

³1.5%~10% of the original computational time reported by PhotoWCT (19)

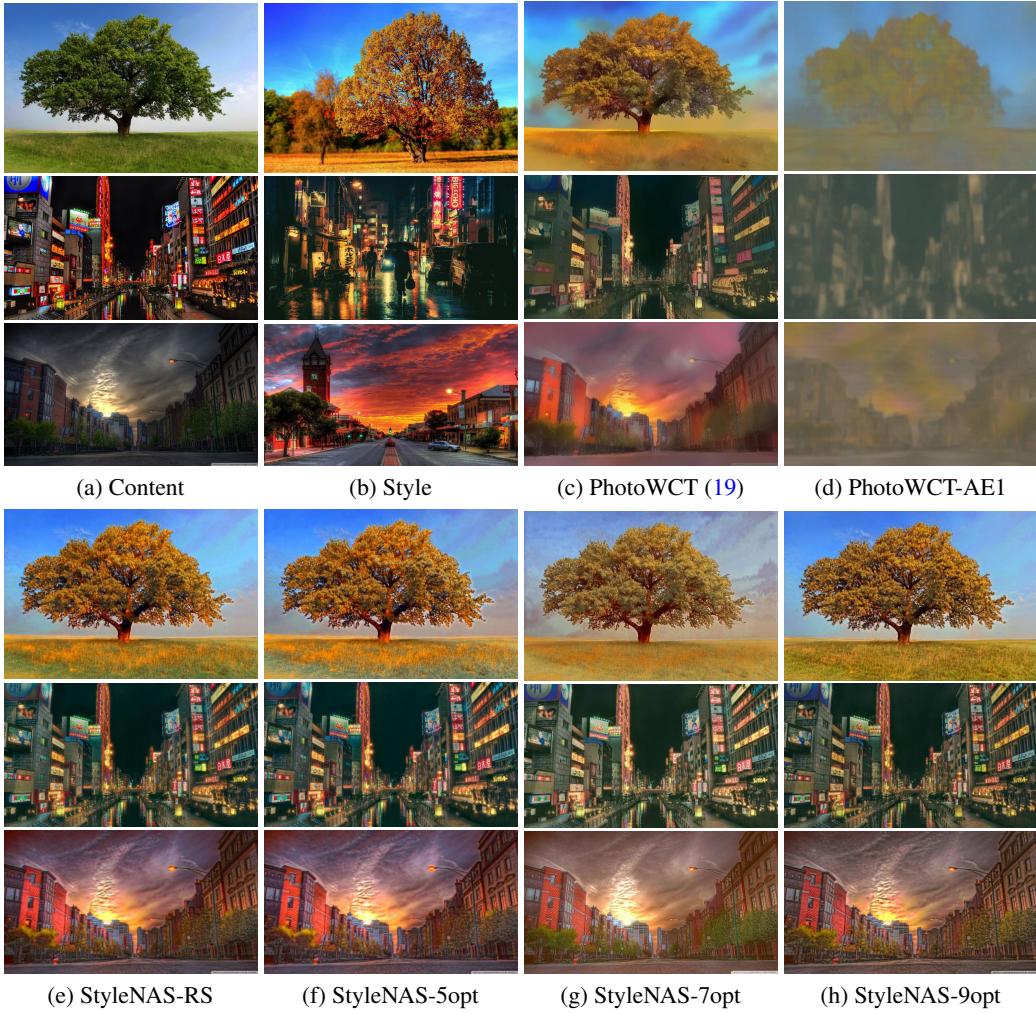


Figure 4: Photorealistic style transfer results comparison against state-of-the-art method and randomly searched architecture. (zoom-in for details.)

4.1 Experiment Setups

To use StyleNAS algorithms, we define a handcrafted network with an architecture drawn from the search space of our StyleNAS as the *Supervisory Oracles*. In the experiments that we reported, the hyper-parameters of StyleNAS are set as $\alpha = 0.8$, $\beta = 0.1$, and $\gamma = 0.1$ to search highly efficient and effective architectures. Through the linear combination of three objective functions introduced in Eq. 3, the StyleNAS algorithm found a group of time-efficient network architectures without stylization effects compromised. The three decreasing lines shown in Figure 5(c) demonstrate that the proposed StyleNAS algorithm can lower the percentage of operators used in θ while ensuring the image qualities i.e., $\mathcal{E}(\theta)$ and $\mathcal{P}(\theta)$ without any compromises. We select three searched networks namely *StyleNAS-5opt*, *StyleNAS-7opt*, and *StyleNAS-9opt* with 5, 7, and 9 operators respectively. The architectures of these networks are shown in Figure 1(b-d).

As a reference, we provide two baseline algorithms to compare with StyleNAS:

PhotoWCT-AE1. A sub-architecture of PhotoWCT using the first AE of PhotoWCT (19) with the WCT modules placed at the bottleneck: i.e., the layers from e1-e5 to d1-d5 in Figure 1(a). The architecture is with a similar scale of parameters as StyleNAS networks.

StyleNAS-RS. A randomly searched network over the same search space of StyleNAS. We randomly draw and evaluates 200 architectures, where StyleNAS-RS is the best architecture with the lowest overall search objective.

Table 1: **Quantitative evaluation results.** A lower FID score means the evaluated method creates images with a more similar style to the reference supervisory oracles/style images. A higher TV score indicates that the measured method preserves more details.

| Method | PhotoWCT | PhotoWCT-AE1 | StyleNAS-RS | StyleNAS-5opt | StyleNAS-7opt | StyleNAS-9opt |
|--------------|----------|--------------|-------------|---------------|---------------|---------------|
| FID-Style ↓ | 180.19 | 469.54 | 208.67 | 183.95 | 173.18 | 172.00 |
| FID-Oracle ↓ | - | - | 142.83 | 65.94 | 50.71 | 45.81 |
| TV score ↑ | 5.11 | 0.43 | 4.09 | 5.73 | 6.14 | 5.83 |

Table 2: **Computation time comparison.**

| Method | PhotoWCT | PhotoWCT-AE1 | StyleNAS-RS | StyleNAS-5opt | StyleNAS-7opt | StyleNAS-9opt |
|------------|----------|--------------|-------------|---------------|---------------|---------------|
| 256 × 128 | 4.38 | 0.83 | 0.30 | 0.05 | 0.07 | 0.47 |
| 512 × 256 | 25.37 | 0.99 | 0.35 | 0.09 | 0.10 | 0.67 |
| 768 × 384 | 64.73 | 1.10 | 0.42 | 0.15 | 0.18 | 0.76 |
| 1024 × 512 | 153.25 | - | 0.52 | 0.23 | 0.29 | 0.91 |

4.2 Experimental Results Comparison

We present the experiment results by comparing the quality of produced images for photorealistic style transfer and the computation time.

4.2.1 Visual Comparisons

In this study, all style transfer networks are evaluated on a dataset consisting of 40 content images and 40 style images, where each content image is transferred into the corresponding style. Figure 4 demonstrates the contrast examples of style transferred images produced by our three searched architectures (i.e., StyleNAS-5opt, StyleNAS-7opt, and StyleNAS-9opt searched by StyleNAS with the handcrafted network as supervisory oracle). We also compare the result with PhotoWCT (19), PhotoWCT-AE1, and StyleNAS-RS. From our visual comparison, we observe that PhotoWCT (19) generates images with quite a lot of details lost (shown in (c)). For example, the grassland in the top image, the text on the advertising boards in the middle image, and the sky in the bottom image are blurred. Results of StyleNAS-RS (shown in (h)) have compromised stylization effects and the generated images are comparably of poor-quality. The StyleNAS-Xops networks (shown in (e-g)) create images with abundant details without compromise of style transfer effects. PhotoWCT-AE1 has a similar time-consumption as the searched models. However, the PhotoWCT-AE1 fails to generate photorealistic images, which demonstrates the effectiveness of the StyleNAS in finding effective networks.

4.2.2 Quantitative Comparison

We compute the following evaluation metrics based on totally all generated images.

FID-Style (12). We compute the Fréchet Inception Distance (FID) (12) score between the reference style images and transferred images by all evaluated methods. As Table 1 (row 1) shows, StyleNAS-opt7/opt9 outperforms all the compared methods with a higher FID score (i.e., better stylization). Note that FID was originally used to validate the image quality for domain adaption and image translation, which is pretty suitable to measure photorealistic stylization effects.

FID-Oracle. We further compute the FID score between the supervisory oracle and the transferred images of searched architectures to demonstrate the effectiveness of StyleNAS against Random Search strategy. Table 1 (row 2) shows that the FID-Oracle score of StyleNAS-Xopt is 2× to 3× lower than StyleNAS-RS, which indicates that StyleNAS models have distinctly better performance than randomly explored networks.

Total Variation (31). We compare the total variation scores of the results by all evaluated methods. As demonstrated in Table 1 (row 3), images generated by StyleNAS-Xopt are of higher total variation scores (i.e., more sharpness and details) than PhotoWCT (19), PhotoWCT-AE1, and StyleNAS-RS.

4.2.3 Computational Time Comparison

We conduct a computing time comparison against the state-of-the-art methods to demonstrate the efficiency of the StyleNAS network architectures. All approaches are tested on the same computing platform which includes an NVIDIA 1080Ti GPU card with 12GB RAM. The time consumption of PhotoWCT (19) is evaluated by running officially released code with default settings. We compare the computing time on content and style images with different resolutions. As Table 2 shows, the StyleNAS-5opt/7opt architectures are almost 300× faster (on 768 × 384 images) than PhotoWCT (19)

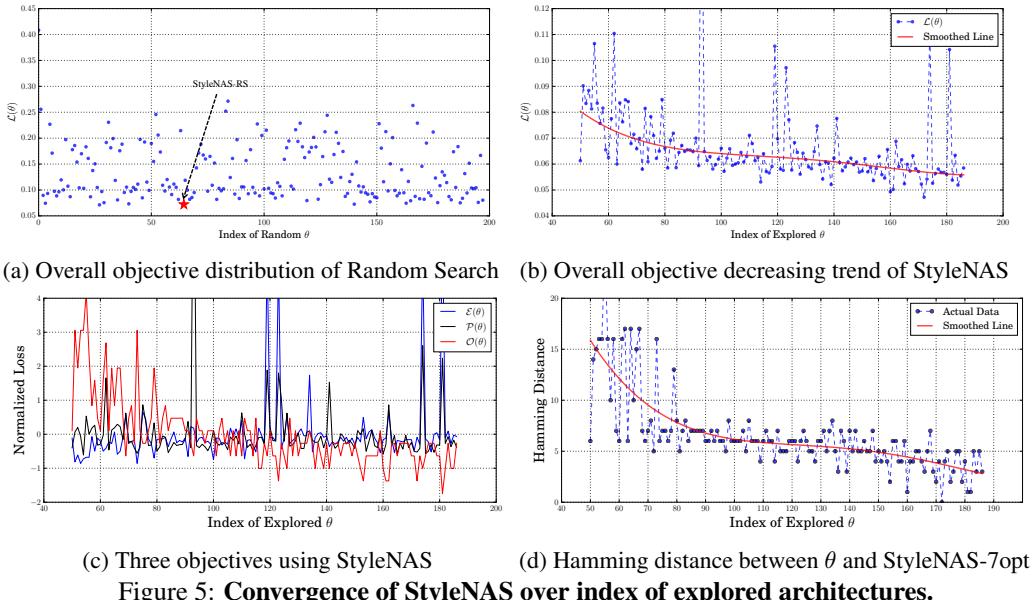


Figure 5: **Convergence of StyleNAS over index of explored architectures.**

method and achieves near-real-time efficiency. Though having nine operators, StyleNAS-9opt model is still $80\times$ faster than PhotoWCT (19) and even PhotoWCT-AE1.

5 Discussion

5.1 Search Effectiveness Analysis of StyleNAS

Our experimental results show that StyleNAS is much more effective than Random Search. In our experiments, the StyleNAS algorithm explores 20 architectures per round. We let StyleNAS search architectures for 7 rounds. A total of 140 architectures were obtained. Among them, 137 architectures were evaluated and the rest 3 architectures failed in training. We also used the Random Search strategy to randomly draw 200 architectures from the search space. We picked up the best architectures from both search methods with the lowest overall objective. The best one from random search, i.e., StyleNAS-RS has the objective function value of 0.0709 while the best one from StyleNAS, i.e., StyleNAS-7opt has the objective function value of 0.0472 (in our study for the objective function value the smaller the better).

We also performed quantitative analysis to compare architectures obtained by StyleNAS and Random Search. In terms of time consumption, StyleNAS-RS spent a significantly longer time than StyleNAS-7opt for the style transfer tasks of all image resolutions evaluated. While the fastest network searched (StyleNAS-5opt) only consumes 16%~40% time of StyleNAS-RS. Though StyleNAS-9opt spent longer time than StyleNAS-RS in our experiments. However, the image quality obtained by StyleNAS-9opt as well as StyleNAS-7opt is much better than StyleNAS-RS, which is demonstrated by Table 1. All in all, StyleNAS-RS cannot outperform StyleNAS-7opt in both quality and complexity wises. Please refer to Tables 1 and 2 for detailed comparisons.

5.2 NAS Convergence Analysis

StyleNAS can converge in terms of both overall search objective and the architectures. Figure 5(b) showed that the overall search objective of StyleNAS decreased over the number of architectures explored. We further broke down the overall objective to its three parts, where Figure 5(c) demonstrated that while the two objectives of image qualities (i.e., reconstruction error and perceptual loss) were ensured at lower levels, the third objective — the numbers of operators decrease in a general sense. The trends of searched networks of StyleNAS demonstrated the size of explored architectures became smaller and smaller while the stylization effects did not compromise.

We found the architectures searched by StyleNAS would be also converged. We took the binary code of StyleNAS-7opt architecture as a reference, and estimated the hamming distance of every explored architecture in the *history set* to the StyleNAS-7opt using binary codes. Figure 5(d) provided the yet first evidence of the convergence in the search space, with simple evolutionary NAS strategies.

6 Conclusions

We have performed an empirical study of Neural Architecture Search to a very important problem of image analysis: that of universal photorealistic style transfer. In our study, we carefully designed the search space and utilized a multi-objective neural architecture search to obtain the first-of-its-kind end-to-end neural networks with improved computational efficiency for style transfer. Our study also identified new characteristics of NAS such as architecture convergence and enhanced search efficiency comparing to random search. In our future work, we plan to extend the work to other generative models such as generative adversarial networks and other low-level vision tasks.

References

- [1] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. *CoRR*, abs/1611.02167, 2016. [3](#)
- [2] Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. Efficient architecture search by network transformation. In *AAAI*, 2018. [3](#)
- [3] Han Cai, Jiacheng Yang, Weinan Zhang, Song Han, and Yong Yu. Path-level network transformation for efficient architecture search. *arXiv preprint arXiv:1806.02639*, 2018. [3](#)
- [4] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *CoRR*, abs/1812.00332, 2018. [3](#)
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: a large-scale hierarchical image database. In *CVPR*, 2009. [4](#), [5](#)
- [6] Oriel Frigo, Neus Sabater, Julie Delon, and Pierre Hellier. Split and match: example-based adaptive patch sampling for unsupervised style transfer. In *CVPR*, 2016. [4](#)
- [7] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. [4](#)
- [8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. [4](#)
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. [2](#)
- [10] Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH*, 1998. [4](#)
- [11] Aaron Hertzmann, Charles E Jacobs, Nuria Oliver, Brian Curless, and David H Salesin. Image analogies. In *SIGGRAPH*, 2001. [4](#)
- [12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017. [7](#)
- [13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. [2](#)
- [14] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. [4](#)
- [15] Ye-Hoon Kim, Bhargava Reddy, Sojung Yun, and Chanwon Seo. Nemo: Neuro-evolution with multiobjective optimization of deep neural network for speed and accuracy. In *ICML 2017 AutoML Workshop*, 2017. [3](#), [4](#), [5](#)
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. [2](#)
- [17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015. [2](#)

- [18] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *NIPS*, 2017. [2](#), [3](#), [4](#)
- [19] Yijun Li, Ming-Yu Liu, Xuetong Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *ECCV*, 2018. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [20] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017. [4](#)
- [21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: common objects in context. In *ECCV*, 2014. [4](#)
- [22] Chenxi Liu, Barret Zoph, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan L. Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. *CoRR*, abs/1712.00559, 2017. [2](#), [3](#)
- [23] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical representations for efficient architecture search. *CoRR*, abs/1711.00436, 2017. [3](#)
- [24] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. *CoRR*, abs/1806.09055, 2018. [2](#), [3](#)
- [25] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *CVPR*, 2017. [2](#), [4](#)
- [26] Renqian Luo, Fei Tian, Tao Qin, and Tie-Yan Liu. Neural architecture optimization. *CoRR*, abs/1808.07233, 2018. [3](#), [4](#)
- [27] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *CoRR*, abs/1802.03268, 2018. [3](#)
- [28] Ori Press, Tomer Galanti, Sagie Benaim, and Lior Wolf. Emerging disentanglement in auto-encoder based unsupervised image content transfer. 2019. [2](#)
- [29] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. *CoRR*, abs/1802.01548, 2018. [3](#)
- [30] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Quoc V. Le, and Alex Kurakin. Large-scale evolution of image classifiers. *CoRR*, abs/1703.01041, 2017. [3](#)
- [31] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. [7](#)
- [32] YiChang Shih, Sylvain Paris, Connelly Barnes, William T Freeman, and Frédo Durand. Style transfer for headshot portraits. *ACM Transactions on Graphics*, 33(4):148, 2014. [4](#)
- [33] Yichang Shih, Sylvain Paris, Frédo Durand, and William T Freeman. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics*, 32(6):200, 2013. [4](#)
- [34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [2](#), [4](#), [5](#)
- [35] Holger Winnemöller, Sven C. Olsen, and Bruce Gooch. Real-time video abstraction. *ACM Transactions on Graphics*, 25(3):1221–1226, 2006. [4](#)
- [36] Saining Xie, Alexander Kirillov, Ross Girshick, and Kaiming He. Exploring randomly wired neural networks for image recognition. *arXiv preprint arXiv:1904.01569*, 2019. [3](#)
- [37] Zhao Zhong, Zichen Yang, Boyang Deng, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Blockqnn: Efficient block-wise neural network architecture generation. *CoRR*, abs/1808.05584, 2018. [3](#)

- [38] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. 2017. [2](#),
[3](#), [4](#)
- [39] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable
architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017. [2](#), [3](#)

Supplementary Material

A Pseudocode of StyleNAS

Algorithm 1 StyleNAS algorithm

Train the *supervisory oracle* (\mathcal{SO}) network;
Set overall search space Θ , revolution cycle C , the population size P , the population/history sets $\Theta^{pop}, \Theta^{history} \leftarrow \emptyset$, generation index $gen \leftarrow 0$;

while $|\Theta^{pop}| < P$ **in parallel do**

- $\theta \leftarrow \text{RANDOMARCHITECTURE}(\Theta);$
- $\theta.loss \leftarrow \mathcal{L}(\theta)$ through training and evaluating a network based on θ and \mathcal{SO} ;
- $\theta.gen \leftarrow gen;$
- $\Theta^{pop} \leftarrow \Theta^{pop} \cup \{\theta\};$
- $\Theta^{history} \leftarrow \Theta^{history} \cup \{\theta\};$

end while

while $|\Theta^{history}| < C$ **do**

- $gen \leftarrow gen + 1;$
- for** $i < P$ **in parallel do**

 - Randomly pickup a *subset* of architectures from Θ^{pop} as $\Delta^{pop} \subseteq \Theta^{pop}$;
 - Set $\theta^{parent} \leftarrow \underset{\theta \in \Delta^{pop}}{\operatorname{argmin}} \theta.loss$ using the architecture in Δ^{pop} with minimal loss;
 - $\theta^{child} \leftarrow \text{MUTATE}(\theta^{parent})$
 - $\theta^{child}.gen \leftarrow gen;$
 - $\theta^{child}.loss \leftarrow \mathcal{L}(\theta^{child})$ through training and evaluating a network based on θ^{child} and \mathcal{SO} ;
 - $\Theta^{pop} \leftarrow \Theta^{pop} \cup \{\theta^{child}\};$
 - $\Theta^{history} \leftarrow \Theta^{history} \cup \{\theta^{child}\};$
 - Set $\theta^{oldest} \leftarrow \underset{\theta \in \Theta^{pop}}{\operatorname{argmin}} \theta.gen$ using the architecture in Θ^{pop} with minimal generation index;
 - $\Theta^{pop} \leftarrow \Theta^{pop} \setminus \{\theta^{oldest}\};$

end for

end while

return $\theta^{best} \leftarrow \underset{\theta \in \Theta^{history}}{\operatorname{argmin}} \theta.loss$ using the architecture in $\Theta^{history}$ with minimal loss;

B More Style Transfer Results

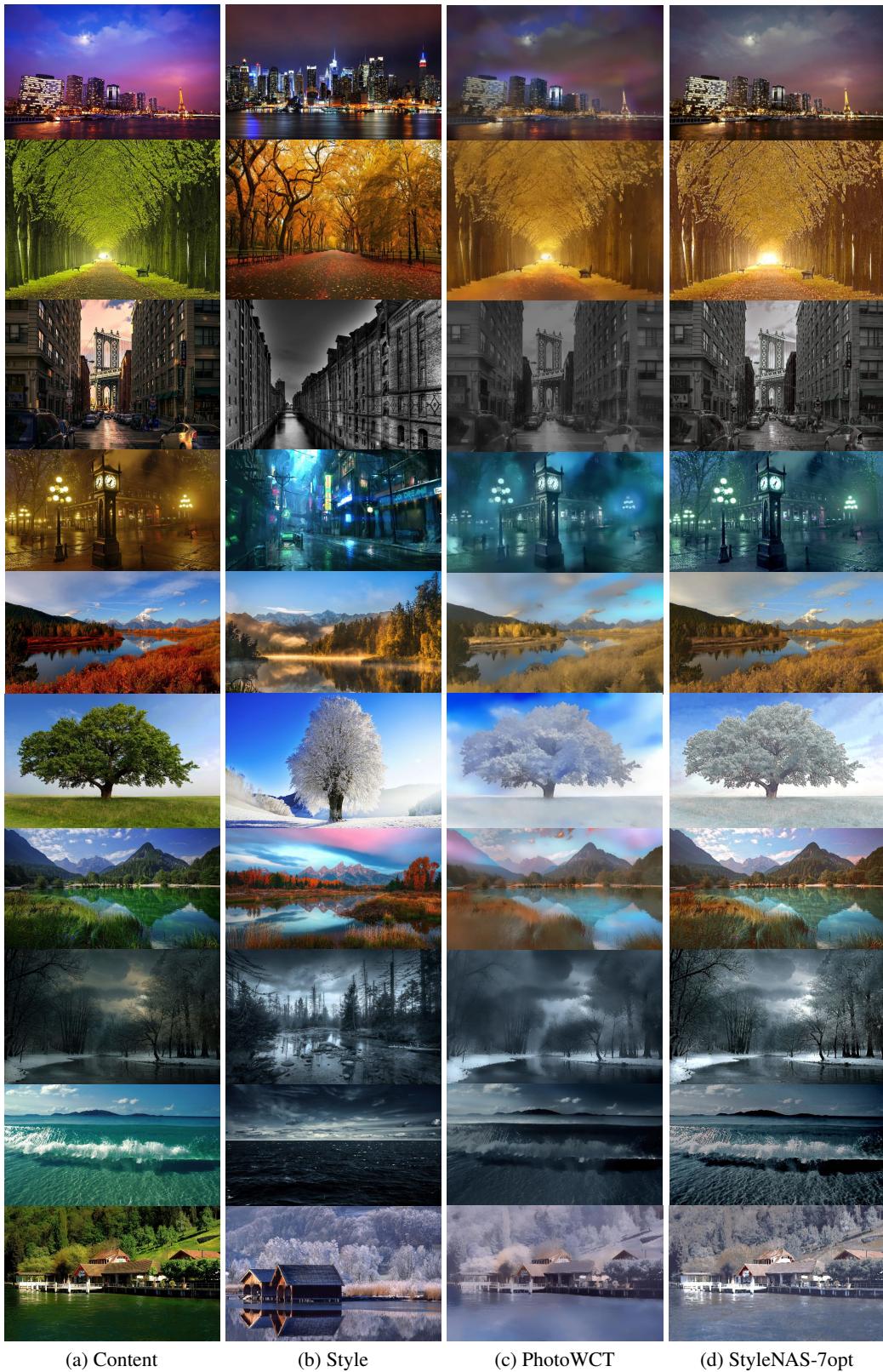


Figure 6: Photorealistic style transfer results comparison.