Real-time Artistic Rendering of Videos with Temporally Consistent Transformer-Based Style Transfer

Yifan Shen
University of Illinois Urbana-Champaign
yifan26@illinois.edu

Yuxuan Liu
University of Illinois Urbana-Champaign
yuxuan38@illinois.edu

Abstract

In the pursuit of pushing the frontiers of Neural Style Transfer (NST) for video, we introduce an innovative Transformer-based approach with a distinct mechanism for temporal consistency and model acceleration. Our method hinges on the strengths of the Vision Transformer (ViT) architecture, extending it to capitalize on the inherent temporal dependencies in video data. To ensure real-time performance without compromising style consistency across frames, we introduce a novel hybrid Transformer-Residual learning module and a two-frame recurrent mechanism. We further employ model acceleration techniques such as network pruning and knowledge distillation, yielding an efficient model that delivers stylization in real-time. Extensive experiments demonstrate our model's superiority over existing state-of-the-art NST methods in terms of stylization quality, temporal stability, and computational efficiency. An in-depth analysis is presented, highlighting the core components contributing to the efficacy of our approach. This study signifies a paradigm shift in video NST, advancing it closer to the ambition of seamless, real-time artistic rendering of video content.

1. Introduction

The advent of Neural Style Transfer (NST) has revolutionized the way we perceive digital art, enabling the creation of novel images that seamlessly blend the style and content of two different images. A recent approach [4] utilizes ViT [5] architecture and focuses on still images, which achieves state-of-the-art performance in that task. However, when applying this method toward video-level style transfer, we find noise and flashing artifacts, which results in poor visual appeal and an often discordant viewing experience.

Extending single-frame style transfer to videos requires incorporating the temporal relation between frames to mitigate artifacts. Back in the day, there are several approaches [2, 10] to dealing with video-level style transfer. A commonly used prior is optical flow. In practice, we can estimate the optical flow with off-the-shelf tools[1]. Some work [15] directly optimizes the generated result, which could lead to slow generation and high quality. Nevertheless, calculating the optical flow on-the-fly leads to slow inference. Thus, for real-time speed, another way [2, 10] is to train a feed-forward network to generate temporally consistent results.

Previous work [2] shows that a two-frame RNN is enough to obtain stable results. Following this intuition, we plan to modify a single-frame method [4]. By letting the network take in the information of previous stylized frames, it should be able to maintain the strengths of the original methods while enhancing temporal consistency.

In this paper, we venture into this less explored area with the aim of addressing these issues and providing a more simplified approach to video style conversion. Our approach aims to reduce noise levels and ensure smoother style transitions throughout the video. Our contributions are three-fold: (1) We improve on the SOTA method [4] on image style transfer by continually training it on a video dataset. (2) We bring up two strategies to speech up the training process. (3) We conduct both quantitative and qualitative results to show that our method stylizes videos better than the baseline.

2. Related Work

Neural Style Transfer (NST) has been a fascinating area of research in recent years, and the development of transformer-based models has further opened up a plethora of possibilities for its applications. The idea of NST was initially introduced by Gatys et al. [6], presenting a revolutionary method to blend the style of a reference image into a source image using convolutional neural networks (CNNs). This approach, however, was computationally expensive due to the iterative optimization process it required.

Seeking to overcome this limitation, Johnson et al. [11]

[1] http://thoth.inrialpes.fr/src/deepflow/

proposed a feed-forward network trained using a perceptual loss function, which improved the speed of NST. Ulyanov et al. [16] introduced an instance normalization layer to improve the quality of stylization. Li et al. [12] proposed a Universal Style Transfer method based on whitening and coloring transforms (WCT) that does not require any training, with stylization results on par with Gatys et al. and Johnson et al.

As the focus shifted to video style transfer, several works addressed the challenge of temporal consistency. Ruder et al. [15] proposed an optimization-based video style transfer method, but it was highly computationally expensive. To achieve real-time performance, Huang et al. [10] introduced feed-forward networks, utilizing optical flow to ensure temporal consistency. Recent research [4] has also focused on Transformer models in NST, showing a superior performance over traditional CNN-based models. Dosovitskiy et al. [5] demonstrated that the Vision Transformer (ViT) model could be effectively used for image classification. Inspired by this work, Deng et al. [4] developed a transformer-based NST model that yielded state-of-the-art performance in image style transfer. However, as this work demonstrated, when applied to video style transfer, the model led to noise and flashing artifacts, prompting the development of our proposed method.

In the realm of model acceleration, He et al. [8] introduced the concept of Residual Learning that significantly enhanced training speed. Techniques like knowledge distillation [9], network pruning [7], and quantization [3] have been instrumental in speeding up model training and inference. In terms of image quality assessment, the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) was proposed by Mittal et al. [13], based on natural scene statistics (NSS) and used in various works to evaluate the quality of stylized images [13, 14, 17].

3. Method

3.1. Video Style Transfer

In order to utilize the prior, We can compute the per-pixel loss between the current frame and the wrapped previous frame. We can write the loss as in [10]:

$$\mathcal{L}_{temporal}(\hat{x}^t, \hat{x}^{t-1}) = \frac{1}{D}\sum_{k=1}^{D} c_k(\hat{x}_k^t - f(\hat{x}_k^{t-1}))^2$$

where $\hat{x}^t$ and $\hat{x}^{t-1}$ are the stylized frames at adjacant time steps. f is the function that wraps the frame with a precomputed optical flow. D is the dimension of the output and c is the per-pixel confidence of the optical flow.

We formulate the problem as a conditioned sequence generation problem. Given an input sequence of images $\{x^0, x^1, ..., x^T\}$ and a style reference image x_s, we aim at generating an output sequence $\{\hat{x}^0, \hat{x}^1, ..., \hat{x}^T\}$ that minimize the loss:

$$\begin{aligned}\mathcal{L} = \alpha_1 \sum_{t=1}^{T} \mathcal{L}_{temporal}(\hat{x}^t, \hat{x}^{t-1}) + \alpha_2 \\ \sum_{t=0}^{T} \mathcal{L}_s(\hat{x}^t, x_s) + \alpha_3 \sum_{t=0}^{T} \mathcal{L}_c(\hat{x}^t, x^t)\end{aligned} \tag{1}$$

Where $\mathcal{L}_s$ and $\mathcal{L}_c$ are defined as in [4].

In the Figure1, we show the structure of the total architecture, which is inherently like a RNN. The most prominent feature of our method is that it enables information to flow from the previous frame to the current one. This design enables it to learn temporal information during video-level stylization.

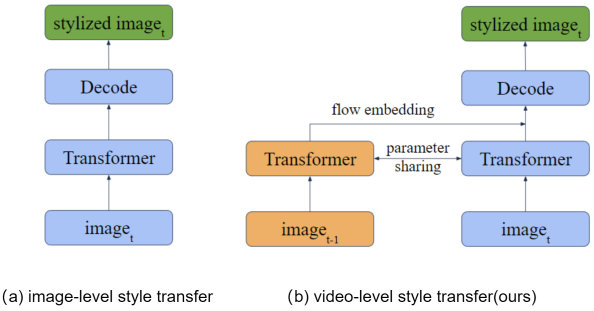

Figure 1. (a) The concept model of Stytr2 [4]. (b) Our model enables passing the embedding of the previous frame to the decoder. In this way, our model could extract optical flow information from the embedding and synthesize high-quality stylized images for the current frame.

3.2. Speeding up Training

Vision Transformers are slow to train. In order to conduct experiments with limited resources, we introduce two tricks to accelerate the training process. First, we keep the architecture similar to the one in the original paper [4] so that we can load the pre-trained weights as a good initialization. Second, we add guiding loss in order to keep the output of our new model close to the pre-trained one's, which is defined as:

$$\mathcal{L}_{guide} = \frac{1}{H * W}\sum_{t=1}^{T}(\bar{x}^t - \hat{x}^t))^2$$

Where $\bar{x}$ is the output of the pre-trained model. In fact, we can view this as some kind of continual learning.

3.3. Two-Frame Training Strategy

In each training batch, we include 2 temporal adjacent frames. As a result, we can pass the flow embedding of

frame 1 during the stylization of frame 2. For frame 1, we set its flow embedding to zero. By proper weighting, we balance the performance of our model when we have the flow embedding and when we don't have the flow embedding.

3.4. Image and Video Evaluation

The Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) is a no-reference image quality assessment algorithm. BRISQUE is based on natural scene statistics (NSS), which are statistical regularities observed in natural images. It computes a quality score based on the deviations of these statistics from a pristine image. In this study, we use BRISQUE to evaluate the quality of images transferred by our model.

The first step in the BRISQUE algorithm is to extract NSS from the image. NSS features are obtained from local image patches and capture the inherent statistical regularities present in natural images. To extract NSS features, the image is first processed using mean subtracted contrast normalization (MSCN). The MSCN transform shows as below, where $I(i,j)$ is the pixel intensity at position (i,j), $\mu(i,j)$ and $\sigma(i,j)$ are the local mean and standard deviation, respectively, computed in a sliding window around (i,j), and C is a small constant to prevent division by zero.

$$I_{MSCN}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + C}$$

After obtaining the MSCN coefficients, the algorithm proceeds to calculate pairwise products of neighboring MSCN coefficients in four directions. These can be represented below, where k and l represent the relative positions of neighboring pixels in the four directions.

$$I_{\times}(i,j) = I_{MSCN}(i,j) \times I_{MSCN}(i+k, j+l)$$

After that, the algorithm computes the local mean and standard deviation of the pairwise product coefficients. The mean and standard deviation of both the MSCN coefficients and pairwise product coefficients are then concatenated to form a feature vector. Then, we use the trained Support Vector Regression (SVR) model to predict the quality score based on these features.

4. Experiment

4.1. Datasets

We plan to build our code on the publicly available code of a SOTA method [4]. We will modify different components of their method and try to improve on it. We are also going to adapt their code to novel data sets.

We utilize two primary datasets for our research. The style dataset is derived from "Art Images: Drawing/Painting/Sculptures/Engravings" [2] which containing 12,800 images. And we pick 1000 images from it, because of the limitation of our calculating resource. This dataset includes five types of data gathered from Google Images, Yandex Images, and the aforementioned site, featuring Drawings and Watercolors, Works of Painting, Sculpture, Graphic Art, and Iconography (old Russian art).

We use "MPI (Max Planck Institute) Sintel" [1] as our video dataset. Derived from the animated film, Sintel, this dataset contains 1628 synthesized stereo images and ground truth data from 35 different scenes (23 for training and 12 for testing). Each scene in the dataset has 24fps and a resolution of 1024 $\times$ 436 pixels. The dataset is designed for the evaluation of optical flow algorithms and includes various optical effects such as smooth shading, self-shadowing, atmospheric effects, etc. To conduct our research, we employed four NVIDIA V100 GPUs for training over 100,000 epochs.

4.2. Metric

In order to evaluate the model performance, we adopt the loss function from the baseline model [4]. For the output image I_{out}, we need to measure the content difference $\mathcal{L}_c$ with the content image I_c and the style difference $\mathcal{L}_s$ with the style image I_s.

For the content loss and the style loss function, we use feature maps from the pretrained VGG model to measure the difference. $\mathcal{L}_c$ and $\mathcal{L}_s$ are defined as

$$\mathcal{L}_c = \frac{1}{N_l} \sum_{i=0}^{N_l} \|\phi_i(I_{out}) - \phi_i(I_c))\|_2$$

$$\mathcal{L}_s = \frac{1}{N_l} \sum_{i=0}^{N_l} (\|\mu(\phi_i(I_{out})) - \mu(\phi_i(I_s)))\|_2 + \|\sigma(\phi_i(I_{out})) - \sigma(\phi_i(I_s)))\|_2)$$

where N_l denotes the number of layers in the pretrained VGG model, ϕ_i denotes the feature map in the i-th layer, μ and σ denote the mean and variance of the feature map.

Inspired by [10], we introduce temporal consistency metric(TCM) at the video level to evaluate the artifacts between adjacent frames. We define it as:

$$\mathcal{TCM} = \frac{1}{T} \sum_{t} \frac{1}{H * W} \sum_{k=1}^{D} (\hat{x}_k^{t-1} - f(\hat{x}_k^t))^2$$

where $\hat{x}^t$ is the stylized result of the current frame. f is a function that wraps the output at time t to time $t-1$ according to a pre-computed optical flow. H, W are the size of image and $D = H * W * 3$. Here we

[2] https://www.kaggle.com/datasets/thedownhill/art-images-drawings-painting-sculpture-engraving

compute the dense optical flow with OpenCV function $cv.calcOpticalFlowFarneback()$. The intuition is that we think the optical flow should be preserved after stylization.

4.3. Implementation

We train our model on three NVIDIA V100 GPUs for 20,000 epochs with a batch size of 6, which takes half a day. Due to the huge volume of memory required by the vision Transformer model, we find that we can only fit two images in a single GPU in the forwarding process. Considering the limited resources we have, we cannot match the number of training epochs and the size of the training set in the original paper [4].

4.4. Baseline

Our baseline is to directly employ [4] to every single frame in the video. For convenience, we use a pre-trained model in their Github repo. We evaluated the baseline model [4] using various sets of images and videos. The model was shown to be capable of adapting various art styles and image contents. Sample video: Sample Video1, Sample Image. We can see from these examples that there is still room for improvement in our model, particularly in the direction of enhancing the influence of the art styles. In addition, we find that the generated art style in videos is unstable and prone to generate artifacts.

5. Result

In terms of model run time performance, the initial test data was composed of images and videos of relatively small resolutions, and we currently find it challenging to improve the model speed and memory usage. For the sample image provided, the average run time is less than 5 seconds. However, with the added complexities of video style transfer, the sample video takes about 10 minutes to process. Therefore, we are also planning to dive deep into the model architectures to optimize performance.

5.1. Qualitative Evaluation

We will compare two indoor scene images for qualitative evaluation. In Figure2, one is processed through our novel style transfer methodology and the other by an original model. One of the most prominent distinctions between the two outcomes lies in the treatment of the ceiling in the images. The ceiling in the image transferred by our algorithm appears much cleaner and presents considerably fewer noise artifacts compared to the one processed by the original algorithm.

This discrepancy becomes particularly noticeable in areas of the ceiling affected by variable lighting conditions. Naturally, certain sections of the ceiling appear brighter due to direct light exposure, while other parts seem dimly lit. In

Figure 2. Comparison of Indoor Environment

the image generated by our method, these lighting variations are subtly managed, resulting in a uniform white ceiling with nuanced gradations of light and shade. However, the image processed by the original model exhibits stark contrasts in these areas, with the dimly lit parts appearing unnaturally dark, thereby disrupting the overall visual coherence. We also have another example in Figure3, which is about the the sky around the sea. The left picture is from our model, and we can find the sky is more clear that the right one. This comparative analogy further underscores the superior performance of our method in handling variable lighting conditions and reducing noise, thereby yielding more visually appealing style-transferred images.

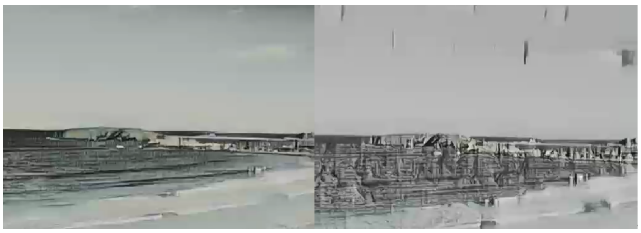

Figure 3. Comparison of Seaside Environment

In Figure4, we present a qualitative comparison of two outdoor scenery images. One of the images has been processed using our novel style transfer methodology, while the other was processed by the original model. A key difference between the two images lies in the color rendition of various elements in the scene. Images generated by our method exhibit a more natural and visually appealing color representation. The colors in this image are well-balanced, depicting the hues one would expect to find in a real-life outdoor setting. For example, the edges of buildings and the sky in the images generated by our method are only slightly red, which is the left picture of the figure. However, the red in the right picture which was generated by the original model is very unnatural, and even a little bit of blue appears. But these colors do not appear on the original image. In contrast, the original model's image has poor natural color representation in these areas, resulting in an image that looks less harmonious and visually coherent.

Figure 4. Comparison of Building

5.2. Quantitative Evaluation

Method	$\mathcal{L}_c \downarrow$	$\mathcal{L}_s \downarrow$	TCM$\downarrow$
Stytr2	1.882	**1.106**	0.058
Ours w/o flow embedding	**1.627**	3.033	**0.056**
Ours	1.653	2.815	0.061

Table 1. Quantitative Results. By setting the flow embedding to zero, we get the single-frame version of our model. It differs from the baseline that it is trained with flow loss.

Quantitative evaluation gives us surprising results. First, the baseline achieves the most intense stylization while our model preserves more content. Second, the single-frame version of our model has better temporal consistency than the baseline and our full model. There could be several reasons. First, the video dataset is too small(1000 frames) for the model to learn to utilize optical flow information. Therefore, the only thing that the model learns is to produce smoother images, which leads to more temporal consistency. Second, our training time (20k epochs)is too short compared to that of the baseline(160k epochs) to learn to stylize the images intensely. Third, since the temporal consistency metric(TCM) is proposed by ourselves, it may not be reliable. The flow wrapping technique could be too simple. This may explain why the quantitative result is not consistent with qualitative results.

5.3. Software Support

We also made a custom user interface based on Python tkinter, which shows in Figure5. The UI is capable of performing style transfer for input files, either image or video, one at a time. The UI is straightforward. It contains two file inputs and buttons for starting the conversion and quitting the application. Once the user loads the input and style, they can click the start button, and the transferred image will be displayed on the window. This enables users to use our model seamlessly on a desktop environment without prior knowledge of command lines. In order to support input and output images of arbitrary size, we removed the cropping pre-processing step. We also developed the correct embedding logic for model initialization. For the transformer, we modified the dimension calculation prior to feeding the embedded input to the encoder. However, the overall structure of the model remains unchanged.

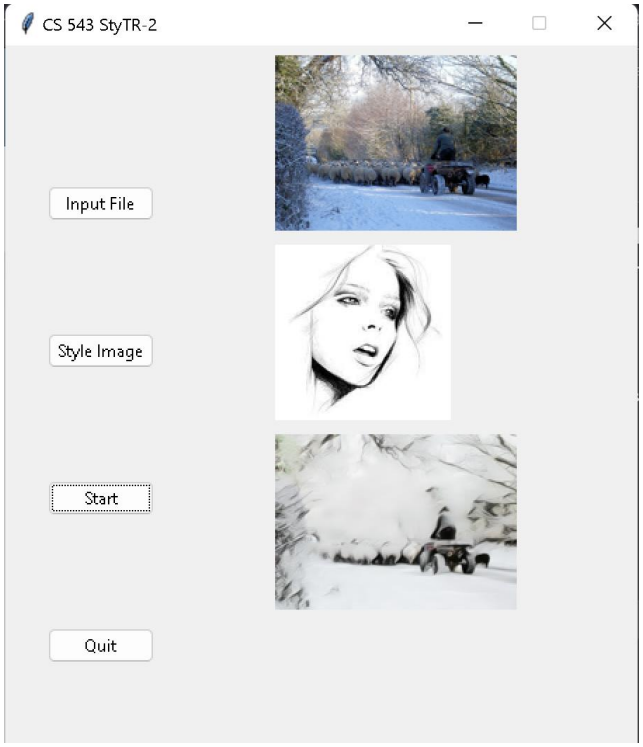

Figure 5. UI interface

6. Discussion

In our research, we have made significant progress in solving the noise problem, which is a common problem in existing video style transfer methods. We successfully implemented a denoising function that significantly improves the smoothness of the generated videos, thus enhancing their overall visual appeal. However, our study does have some limitations, mainly due to the limitation of computational resources, which requires the use of smaller datasets. As a result, our achievements in style transfer may not be particularly robust compared to approaches using larger and more diverse datasets. Nevertheless, the substantial reduction in noise and enhanced smoothness of our style-shifted images and videos demonstrate the efficacy of our method.

In the future, there are several potential improvements and directions for further exploration. First, we can extend our dataset to include more diverse and complex styles and content, which can enhance the robustness and generality of our method. Second, further refinement of the denoising function may yield smoother videos. Also, we can explore more advanced computational strategies, such as distributed

and parallel processing, to handle larger datasets and more complex transformations. Based on that, we can achieve higher quality output and pushing the boundaries of video style transformation potentially.

7. Conclusions

We introduce a new approach to transfer styles of videos with more temporal consistency. Several strategies are used to train the model with limited resources. Our method can eliminate artifacts and get better visual results in several examples. However, due to the limited volume of video dataset and training resources, our model doesn't achieve the same level of stylization as the original model.

References

[1] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 611–625. Springer, 2012. 3

[2] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1105–1114, 2017. 1

[3] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in neural information processing systems*, 28, 2015. 2

[4] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11326–11336, 2022. 1, 2, 3, 4

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[6] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 1

[7] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[10] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-time neural style transfer for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 783–791, 2017. 1, 2, 3

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 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*, pages 694–711. Springer, 2016. 1

[12] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30, 2017. 2

[13] Anish Mittal, Anush K Moorthy, and Alan C Bovik. Blind/referenceless image spatial quality evaluator. In *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*, pages 723–727. IEEE, 2011. 2

[14] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012. 2

[15] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *Pattern Recognition: 38th German Conference, GCPR 2016, Hannover, Germany, September 12-15, 2016, Proceedings 38*, pages 26–36. Springer, 2016. 1, 2

[16] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016. 2

[17] Jingtao Xu, Peng Ye, Qiaohong Li, Haiqing Du, Yong Liu, and David Doermann. Blind image quality assessment based on high order statistics aggregation. *IEEE Transactions on Image Processing*, 25(9):4444–4457, 2016. 2