Real-time Localized Photorealistic Video Style Transferr – Supplementary Material

Xide Xia*
Boston University
xidexia@bu.edu

Tianfan Xue
Google Research
tianfan@google.com

Wei-sheng Lai
Google
wslai@google.com

Zheng Sun
Google Research
zhengs@google.com

Abby Chang
Google
abbychang@google.com

Brian Kulis
Boston University
bkulis@bu.edu

Jiawen Chen
Google Research
jiawen@google.com

1. Algorithm Details

1.1. Network architectures

We provide the architectural details of our mask enhancement network, guidance learner, spatiotemporal feature transfer, and grid learner in Table 1. As shown in Figure 1, we extract the VGG-19 features from four scales (CONV1_1, CONV2_1, CONV3_1, and CONV4_1) and match the multi-resolution feature statistics via three splatting blocks [9]. Note we remove the global scene summary path in Xia et al. [9] as we aim to transfer style between local regions, where the global features will not help. In each splatting block, the weights of the first and the last convolutional layers are shared for the content and style feature paths. The output of each ST-AdaIN layer is propagated to the next frame for blending the temporal information. After the feature transferring, we apply two convolutional layers to predict the bilateral grids.

1.2. Soft grid mask

In Algorithm 1, we provide the detailed steps to compute the soft grid mask for blending the foreground and background grids.

Algorithm 1: Compute Soft Grid Mask

Input: Learned guide map $\mathbf{z}$, pixel mask $\mathbf{M_{pxl}}$, image size (w, h), grid size (W, H, D)
Output: Soft grid mask $\mathbf{M_{grid}}$

Initialize grid mask: $M_{grid} = zeros(W, H, D)$;
$\mathbf{z}_D = floor((\mathbf{z} \cdot \mathbf{M_{pxl}}) \times D)$;
$s_w, s_h = w/W, h/H$;
for $x \leftarrow 1$ *to* W *and* $y \leftarrow 1$ *to* H **do**
  $patch = \mathbf{z}_D[x \times s_w : (x+1) \times s_w, h \times s_h : (y+1) \times y_h]$;
  $M_{grid}[x, y, :] \leftarrow sum(patch > 0)$;
  for $d \leftarrow 1$ *to* D **do**
    if d *in* $patch$ **then**
      $M_{grid}[w, h, d] \leftarrow sum(patch == d)$;
    end
  end
end
Normalize grid mask: $M_{grid} \leftarrow M_{grid}/(s_w \times s_h)$;

1.3. Implementation details

We implement our model in Tensorflow [1]. For training, we use the training set from DAVIS 2017 [8] which contains 60 videos and each video has 70 frames in average. We optimize the model using Adam [3] with hyperparameters $\alpha = 10^{-4}, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$, and a batch size of 2 video clips. At each iteration, we take five consecutive frames as a clip and randomly crop the original frames into a resolution of 256×256. To train the mask enhancement network, we generate noisy input masks by applying erosion and dilation with random kernel sizes to the ground-truth object masks. Since mask enhancement is a largely orthogonal task, we first pre-train this module for 20000 iterations and freeze the weights. We then train the remainder of the network end-to-end for 90000 iterations. By limiting the "full-res" input to also be 256×256, we can significantly reduce training time, which takes about two days on a single NVIDIA Tesla V100 GPU with 16 GB RAM. At inference time, the trained model can be applied to arbitrary input resolution, because of the usage of bilateral grid.

2. Grid Sub-sampling

The proposed method can be further sped up by subsampling the bilateral grids in the temporal domain. As the nearby frames typically have similar color, the bilateral grids can also be shared to render the stylized result.

	Mask Enhancement			Guidance Learner		Spatiotemporal Feature Transfer											Grid Prediction	
	M^1	M^2	M^3	G^1	G^2	S_1^1	S_1^2	S_1^3	S_2^1	S_2^2	S_2^3	S_3^1	S_3^2	S_3^3	L^1	L^2	F	Γ
type	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv	conv
stride	1	1	1	1	1	2	1	1	2	1	1	2	1	1	1	1	1	1
kernel size	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	1
spatial size	256	256	256	256	256	128	128	128	64	64	64	32	32	32	16	16	16	16
channels	16	8	1	16	1	8	8	8	16	16	16	32	32	32	64	64	64	96
activation	-	-	sigmoid	-	sigmoid	relu	relu	relu	relu	relu	relu	relu	relu	relu	relu	relu	relu	-

Table 1: Detailed configuration of each module.

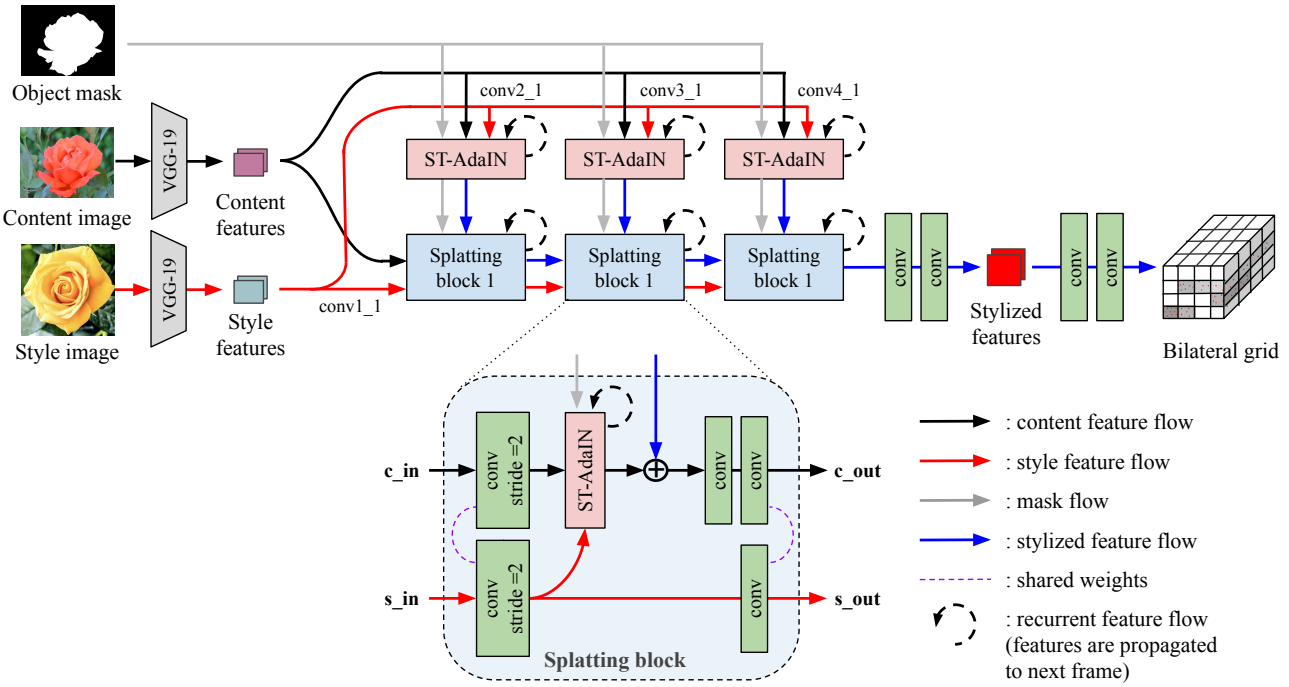


Figure 1: Detailed architecture of our spatiotemporal feature transfer and grid prediction layers.

Specifically, we can generate the bilateral grid for every r frames and estimate the intermediate grids by a linear interpolation, as shown in Figure 2(a). In Figure 2(c) and (d), we show that our method can still render high-quality results up to a sub-sampling rate of $r = 8$. For $r = 16$, the estimated grid may have a larger spatial mismatch with the content, resulting in undesired visual artifacts. Such a temporal sub-sampling strategy is suitable for the proposed method and Xia et al. [9]. Other approaches, e.g., WCT^2 [10], is not able to generate reasonable results by simply interpolating the output frames, as shown in Figure 2(b).

By utilizing such a grid sub-sampling strategy, we can reduce the computational cost and speed up the processing time during inference. As shown in Table 2, our model with sub-sampling rate $r = 8$ is about $8\times$ faster at four different image resolutions. The video comparisons are provided in the supplementary videos *grid_sampling.mp4* in the **video_mp4** folder.

Image Size	512 × 512	1024 × 1024	2000 × 2000	3000 × 4000
Lai et al. [4]	0.0024s	0.0031s	0.0073s	OOM
LST* [5]	0.2753s	0.8365s	OOM	OOM
PhotoWCT* [6]	0.6366s	1.5185s	OOM	OOM
WCT^2* [10]	3.8599s	6.1375s	OOM	OOM
Xia et al.* [9]	0.0058s	0.0068s	0.0117s	OOM
Ours ($r = 1$)	0.0378s	0.0380s	0.0414s	0.0464s
Ours ($r = 8$)	0.0048s	0.0049s	0.0052s	0.0058s

Table 2: Execution time. An asterisk denotes that the technique of Lai et al. [4] was applied to improving temporal stability. OOM indicates out of memory.

3. More Qualitative Comparisons

3.1. Effect of TC-AdaIN

Figure 3 shows a visual comparison, where the results without the TC-AdaIN have a visible color shift. We can see

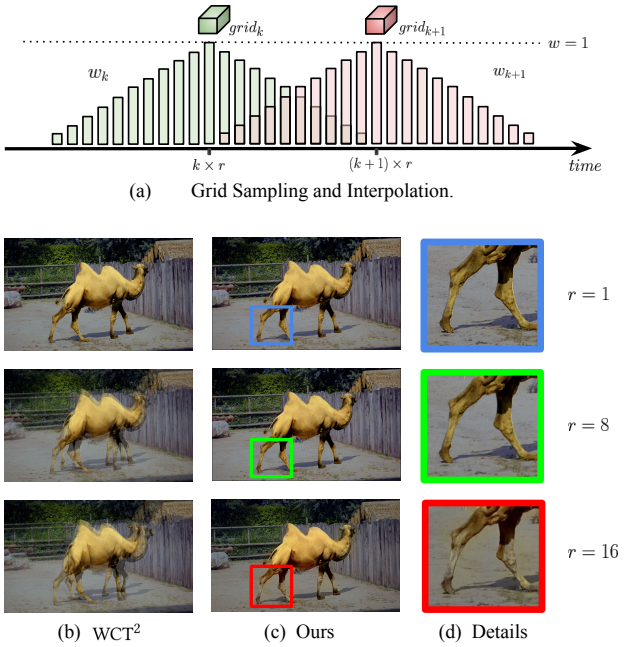


Figure 2: Our method can be sped up by subsampling the bilateral grids in the temporal space while achieving similar visual quality.

Figure 3: **Effect of TC-AdaIN.** Without propagating intermediate features with TC-AdaIN, the stylized outputs have a significant color shifting within 8 frames in this example.

the stylized outputs have a significant color shifting without propagating intermediate features with TC-AdaIN.

3.2. Mask refinement

While our grid-space blending can handle imperfect input masks, visible artifacts may remain if the masks are too noisy, as shown in Figure 4(b) and (e). Our mask enhancement network significantly improves mask boundaries (Figure 4(c)), and our rendered result (Figure 4(f)) is visually comparable to a rendering using the ground truth mask (Figure 4(g)).

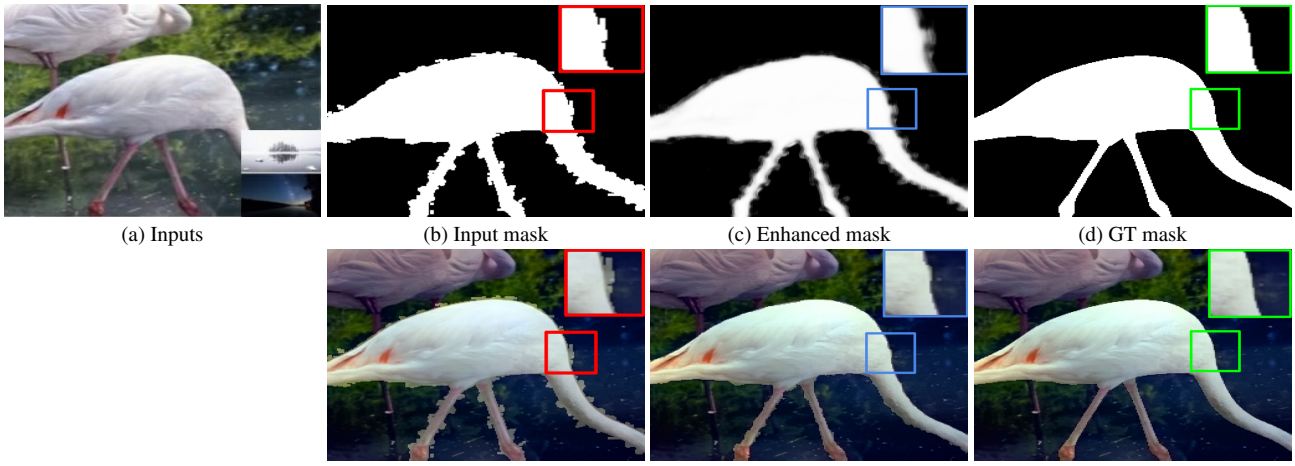
3.3. Anti-distortion module

Recent photorealistic style transfer methods [6, 5, 10] are based on encoder-decoder architecture, which often cause spatial distortions or unrealistic visual artifacts when reconstructing an image from the low-resolution deep features. Therefore, extra smoothing steps or modules are required to minimize those spatial distortions. PhotoWCT [6] optimizes a matting affinity to ensures spatially consistent stylization. Similarly, LST [5] applies a spatial propagation network (SPN) [7] on the reconstructed images to smooth the results. On other other hand, WCT^2 [10] replaced max-pooling layers with wavelet pooling where the high frequency components are skipped to the decoder directly so that all edges and corners are preserved.

Here, we show visual comparisons to existing methods with and without applying their anti-distortion modules. Figure 5 shows the comparison with PhotoWCT. Although the smoothing step helps reduce local artifacts, the smoothed result becomes blurry and hazy. In contrast, our result preserves all the edges and image structures well. In Figure 6, the result of LST without applying SPN has severe spatial distortions. The SPN recovers some image details but cannot suppress all the distortions, resulting in an unrealistic result. Figure 7 shows the results of WCT^2 with and without the skip connections for high-frequency components. It is clear that the skip connections helps bring back the image details and preserve the photorealism, showing comparable results to the proposed method. More video comparisons are provided in the supplementary videos $visual\_comparison.mp4$ in the **video_mp4** folder.

3.4. High-resolution style transfer

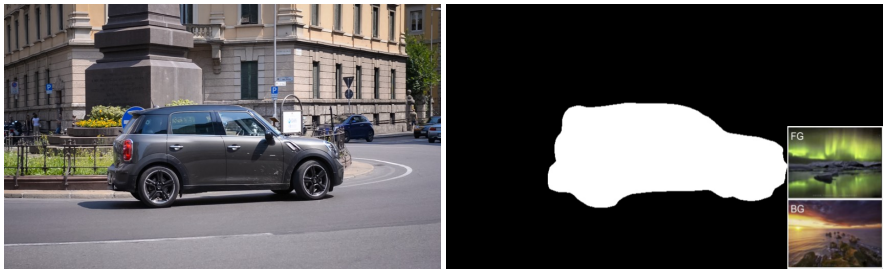
Although the proposed method is efficient when processing high-resolution (e.g., 2000×2000) videos, extracting the object segmentation masks could be computationally expensive. For example, it takes 2.61 seconds for a segmentation model, DeepLab [2], to process a single 1920×1080 frame. Therefore, we aim to understand the feasibility of applying low-resolution segmentation masks for high-resolution style transfer. First, we compute the object masks from a low-resolution (256×256) input frame, which takes only 0.47 seconds for DeepLab. Then, we resize the masks to 1920×1080 through nearest neighbor interpolation for transferring styles. We compare the proposed method with WCT^2 in Figure 8. The results from WCT^2 have clear visual artifacts on the object boundaries as the upscaled masks are noisy. On the other hand, our method is able to generate high-quality results, which are comparable to the ones produced by using the ground-truth object masks. We demonstrates that our model performs well on high-resolution videos even if the segmentation masks is extracted from a low-resolution space. Please refer to the supplementary videos $HD\_video\_w\_upscaled\_mask\_*.mp4$



(a) Inputs (b) Input mask (c) Enhanced mask (d) GT mask

(e) Result with input mask (f) Result with enhanced mask (g) Result with GT mask

Figure 4: Effect of mask enhancement. When the input object mask is too noisy, the stylized results may still contain visible artifacts on the boundaries. Mask enhancement lets us render high-quality boundaries visually comparable to the result using the ground truth mask.



Content Mask and styles

PhotoWCT [6] w/o spatial smoothing PhotoWCT [6] w/ spatial smoothing Ours

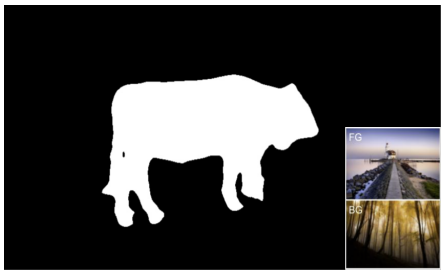
Figure 5: Visual comparison with PhotoWCT [6].

in the **video_mp4** folder for video comparisons.

4. Failure Cases

Our method requires reliable image-space statistics and can cope with modest amounts of segmentation noise. However, it may fail when the selected areas are too small (e.g., a textureless region with a single color). On the other hand, when the foreground objects cannot be detected properly due to occlusion or reflection, our method may not be able to separate the styles of foreground and background very well, as shown in Figure 9.

Content



Mask and styles

LST [5] w/o SPN

LST [5] w/ SPN

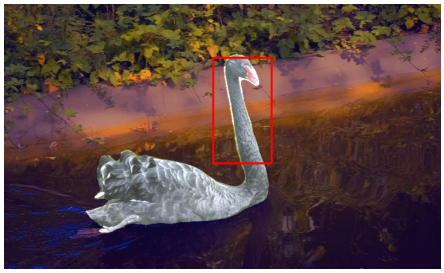
Ours

Figure 6: Visual comparison with LST [5].

Content

Mask and styles

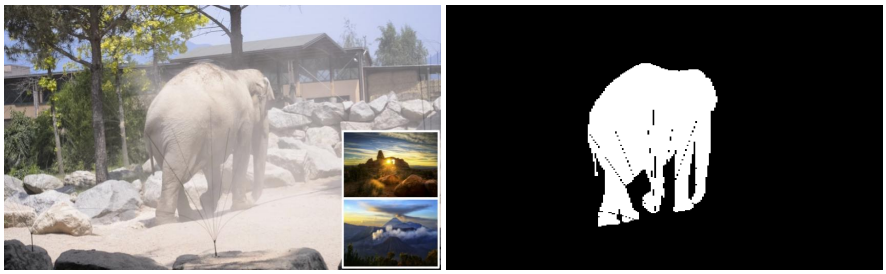
WCT^2 w/o high-freq skips



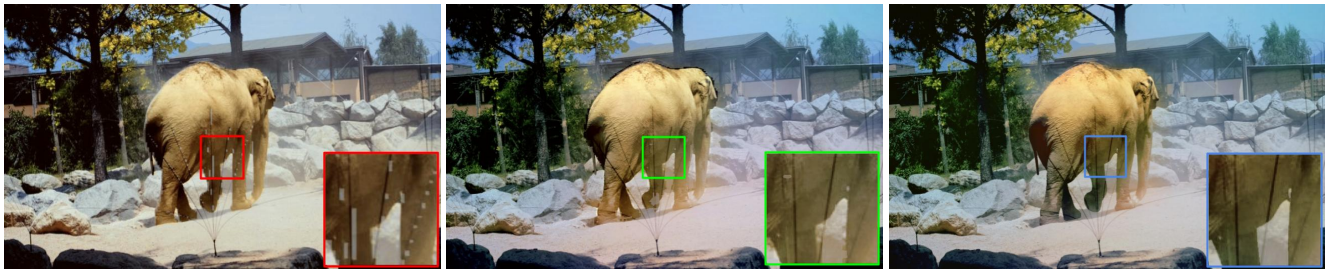
WCT^2 w/ high-freq skips

Ours

Figure 7: Visual comparison with WCT^2.

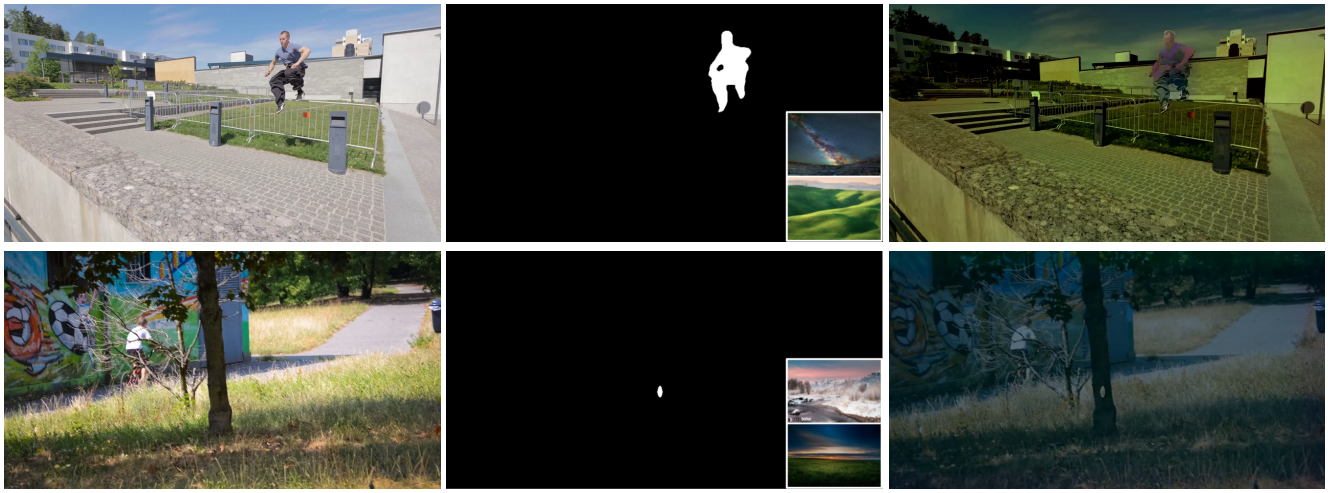


Content and styles | Up-scaled mask



WCT^2 with up-scaled mask | Ours with upscaled mask | Ours with GT mask

Figure 8: High-resolution video stylization using low-resolution masks.



Content | Mask and styles | Our result

Figure 9: Failure cases.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[4] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018.

[5] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *CVPR*, 2019.

[6] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *ECCV*, 2018.

[7] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *NIPS*, 2017.

[8] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv:1704.00675*, 2017.

[9] Xide Xia, Meng Zhang, Tianfan Xue, Zheng Sun, Hui Fang, Brian Kulis, and Jiawen Chen. Joint bilateral learning for real-time universal photorealistic style transfer. In *ECCV*, 2020.

[10] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *ICCV*, 2019.