(Appendix) Smooth-Swap: A Simple Enhancement for Face-Swapping with Smoothness

Jiseob Kim[1,2], Jihoon Lee[2], Byoung-Tak Zhang[1]
[1]Seoul National University, [2]Kakao Brain
jkim@bi.snu.ac.kr, jihoonlee.in@gmail.com, btzhang@bi.snu.ac.kr

Contents

List of Figures

A. Architecture Details

A.1. Identity Embedding Model: f^*_{emb}

Our identity embedding model is based on ResNet-50 [3], where we use a different head for contrastive learning as seen in Fig. A1. Note the *UnitNorm* in the final layer makes z_{src} to be unit-length ($\|z_{src}\| = 1$). The network invovles total 32.3M parameters.

A.2. Swap-Image Generator: f_{gen}

Our generator architecture is mostly the same as NCSN++ [8] except for the following three differences (as described in the main manuscript, Sec. 4.2): 1) we use half as many channels, 2) we use the identity embedding instead of the time embedding, and 3) we add an input-to-output skip connection. Fig. A1 (a) shows the detailed structure with dimensional information. The network involves total 9.8M parameters.

Up/Down Sampling & Skip-Connections Note in each of the outer block containing multiple ResBlocks, the first ResBlock handles upsampling or downsampling (except for the ResBlock x5, where the second ResBlock handles upsampling). There are 13 skip connections in total ($13 = 3 \times 4 + 1$; $+1$ is the input-to-output skip), where the input to each of the ResBlock in the encoder part (before the Attention Block) is handed over to the decoder part (after the Attention Block). On the decoder side, the first three ResBlocks of each outer block get the skip-connections (except for the ResBlock x5, where the second through the fourth get the skip-connections).

Details on the ResBlocks of the Generator We describe some essential details of the *ResBlocks* of the generator here. The complete information can be found in [8].

The overall structure of ResBlock is not much different from the conventional design [3]. However, as shown in Fig. A1 (b), a structure for conditioning on the identity embedding vector z_{src} is added (similar to [1]). The conditioning is done by 1) projecting z_{src} onto a c_{out}-dimensional vector, 2) spatially broadcasting the result, and 3) adding it to the intermediate output of the original path (c_{out} is the number of output channels of the current block). When upsampling or downsampling is used, the optional components (denoted by yellow and dash-dotted outline) are also computed.

Throughput and FLOPs at Inference Time Smooth-Swap generator has much higher FLOPs (in MACs) than HifiFace [9] (214.47G to 102.39G). However, it shows far better throughput (42.96 fps) than others (SimSwap [2]: 31.17, HifiFace [9]: 25.29; FaceShifter [6]: 22.34)[1]. We speculate this is due to the simple and homogeneous architecture, which is advantageous for speed-up with GPU computing.

[1]Test settings and values of other models are adopted from [9]

A.3. Discrminiator: f_{dis}

We use the same discriminator as the one used in StyleGAN2 [5]. The network involves total 28.9M parameters.

B. More Image Samples from Smooth-Swap

We show extended sets of swapped-image samples from our Smooth-Swap model. The following three figures, Fig. A3, A4, and A6 present the results of the same experiments as Fig. 4, 5, and 6 in the main manuscript, but with different source and target pairs. Fig. A5 shows the results for out-of-distribution cases, where oil paintings (Metfaces dataset [4]) are used for swapping. Although the model is never trained on such images, the results are of decent quality, reflecting the characteristics of the source and the target with shape change.

C. Extreme Cases and Limitations

We note that Smooth-Swap can fail when a target image involves occlusion or an extreme pose as shown in Fig. A2. However, we believe each case can be handled by post-processing (e.g., HEAR-Net of [6]) and supplying more extreme-pose examples for training.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations*, 2019. 1

[2] Renwang Chen, Xuanhong Chen, B. Ni, and Yanhao Ge. SimSwap: An Efficient Framework For High Fidelity Face Swapping. *ACM Multimedia*, 2020. 1

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. 1

[4] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training Generative Adversarial Networks with Limited Data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020. 2, 6

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. StyleGAN2. 2

[6] Lingzhi Li, Jianmin Bao, Hao Yang, Dong Chen, and Fang Wen. FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping. *arXiv:1912.13457 [cs]*, Dec. 2019. 1, 2

[7] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. FaceForensics++: Learning to Detect Manipulated Facial Images. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1–11, Seoul, Korea (South), Oct. 2019. IEEE. 4

[8] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021. 1

[9] Yuhan Wang, Xu Chen, Junwei Zhu, Wenqing Chu, Ying Tai, Chengjie Wang, Jilin Li, Yongjian Wu, Feiyue Huang, and Rongrong Ji. HifiFace: 3D Shape and Semantic Prior Guided High Fidelity Face Swapping. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1136–1142. International Joint Conferences on Artificial Intelligence Organization, Aug. 2021. 1

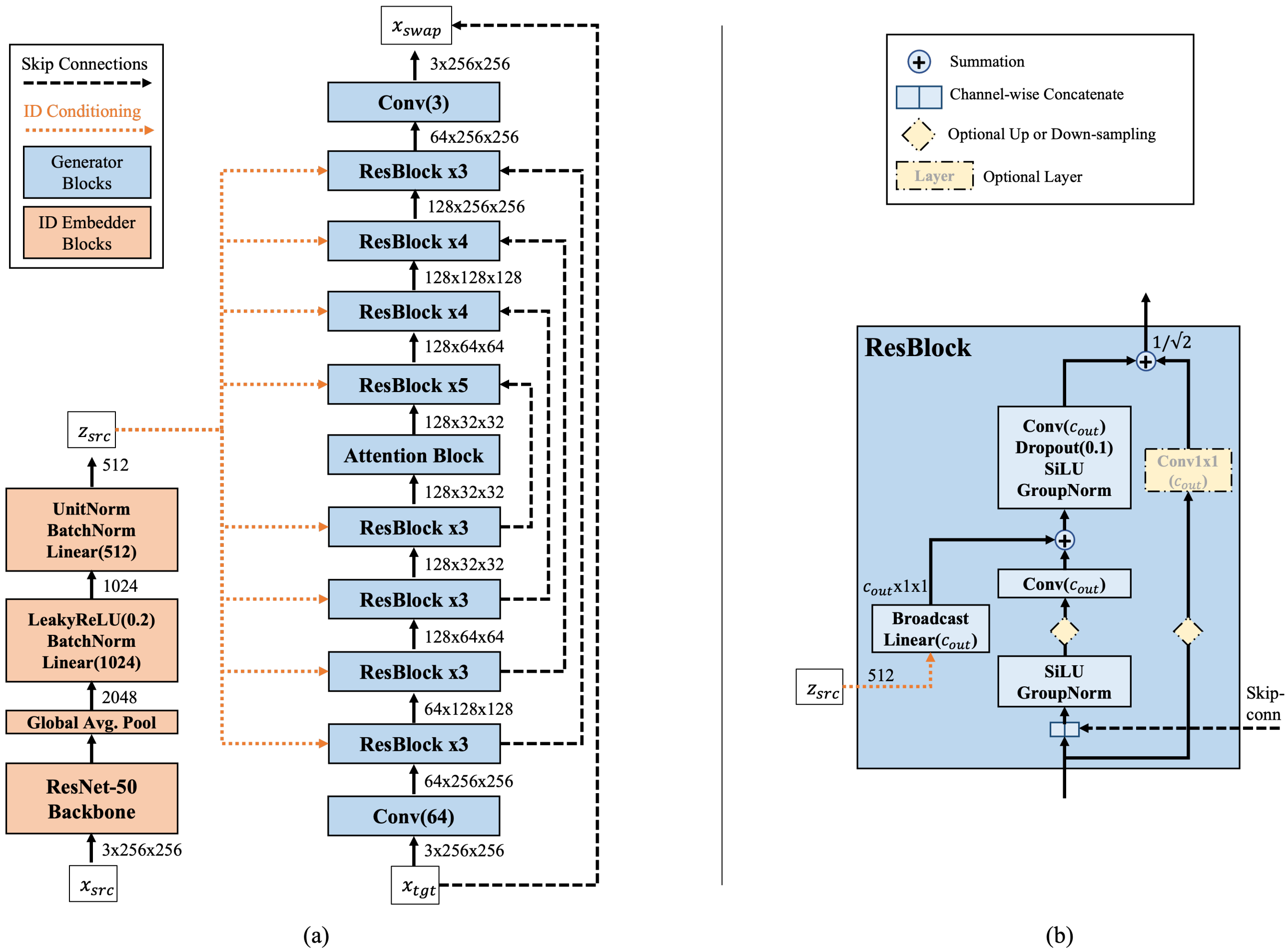

Figure A1. Detailed architecture of our Smooth-Swap model; both the identity embedder and the generator are shown. The intermediate feature-map dimensions are written in the order of (channels $\times$ height $\times$ width). 'ResBlock x4' in (a) denotes that there are four residual sub-blocks connected sequentially; the structure of the sub-block (i.e., ResBlock) is detailed in (b). Note the multi-line text inside blocks should be read from the bottom (e.g., Linear(512) to BatchNorm to UnitNorm for the last embedder block). See Sec. A.2.

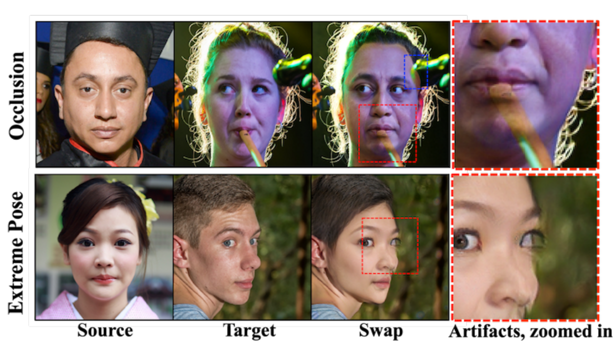

Figure A2. Some failure cases of Smooth-Swap. See Sec. C

Figure A3. Comparison of the face-swapping results of various models on the FaceForensics++ dataset [7] (extension of Fig. 4 in the main manuscript)

Figure A4. More results of Smooth-Swap on the FFHQ test split (extension of Fig. 5 in the main manuscript). Active change of identity is observed. However, in some cases where the source and the target have largely different face shapes (e.g., a child in the rightmost column in the lower-right block), artifacts are noticed. In real-world applications, such cases can be avoided by choosing the swapping pairs from a similar age range.

Figure A5. Results of Smooth-Swap across the FFHQ test split and Metfaces [4]. Even though the model is not trained on the oil paintings of Metfaces, it can still produce swapped images with a decent quality

Figure A6. More face swapping results of Smooth-Swap on wild images (extension of Fig. 6 in the main manuscript).