Supplementary Material: MorphGAN: One-Shot Face Synthesis GAN for Detecting Recognition Bias

Nataniel Ruiz[‡1]
nruiz9@bu.edu
Barry-John Theobald[2]
barryjohn_theobald@apple.com
Anurag Ranjan[2]
anuragr@apple.com
Ahmed Hussein Abdelaziz[2]
hussenabdelaziz@apple.com
Nicholas Apostoloff[2]
napostoloff@apple.com

[1] Boston University
Boston
MA, USA
[2] Apple
Cupertino
CA, USA

A.1. Network Architecture

We use a similar architecture to the generator and style encoder in StarGAN-v2 [1], but modify it to incorporate our shape renders. We design our own global discriminator D_1 and patch discriminator D_2, as described in the following sections.

Generator We use a generator with four down-sampling blocks, four residual blocks and four up-sampling blocks as used by StarGAN-v2 [1]. All blocks have residual connections. Instance Normalization (IN) [5] is used for the down-sampling blocks as well as the first two residual blocks. Adaptive Instance Normalization (AdaIN) [3] blocks are used for the last two residual blocks and for the up-sampling blocks. The AdaIN layers take a style encoding as input. The main difference for our generator is that it takes the reference image and the concatenated target shape render as input, in the form of a 256x256x6 tensor.

Style Encoder We use a similar style encoder to StarGAN v2 [1], except we only use one output branch. The input to the network is a face image. The network itself is composed of an initial 1x1 convolutional layer, followed by six down-sampling residual blocks, a Leaky ReLU, a 4x4 convolutional layer that maps the feature map to a 1x1x512 vector, another Leaky ReLU and a final linear layer.

Global Discriminator We use two branches – an image/render branch and a style branch. The image/render branch has six pre-activation residual blocks with leaky ReLUs using the same layout as the style encoder. The style branch has four linear layers [[1024, 512], [512,

© 2020. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.
‡ This work was part of Nataniel Ruiz's internship at Apple.

256], [256, 128], and [128,32]] all using ReLU activations. Finally, after concatenation of the output features from the image/render branch and the style branch, there is a final branch with three linear layers [[64, 32], [32, 16], and [16, 1]], mapping to a final prediction. The Global discriminator takes in a face image (real or fake), a shape render and a style vector as input.

Patch Discriminator Our patch discriminator only takes in the face image (real or fake) as an input. We use four pre-activation residual blocks in the same fashion as the image/render branch of our global discriminator. The final patch sizes are 8x8.

A.2. Training Details

We use the following values for the full loss presented in the main paper: $\lambda_{\text{VGG}} = 10, \lambda_{\text{VGGFace}} = 10, \lambda_{\text{cyc,VGGFace}} = 1, \lambda_{\text{cyc,VGG}} = 1, \lambda_{\text{sty,ref}} = 1, \lambda_{\text{sty,tgt}} = 1$.

We train the network for 30 epochs on our dataset of 13,000 videos (corresponding to approximately 1M image pairs). We use a batch size of 128, and Adam [4] optimizers with a learning rate of $1e-4$, weight decay $1e-4$, $\beta_1 = 0.0$ and $\beta_2 = 0.99$.

Balancing Pose in Batches We pre-compute pose differences between image pairs in videos, and bin pose differences between the reference image and the target image into the following bins: [[-180 °, -30 °], [-30 °, -15 °], [-15 °, -5 °], [-5 °, 5 °], [5 °, 30 °], [30 °, 180 °]]. We then balance our batches with an equal number of samples from each bin to train with variation in pose differences between reference and target images. This improves the pose fidelity of MorphGAN as well as alleviate artifacts in the rendered output when pose changes are large.

A.3. Qualitative Results

We show additional comparative qualitative results of our one-shot face synthesis model. In Figures 1, 2, 3, 4, 5, we show comparisons between MorphGAN, an unofficial open-source implementation of Zakharov et al. [7] with pre-trained weights, and the official implementation of X2Face [6]. Input images for Zakharov et al. [7] are zoomed out and padded to be within their training domain. We show five different expression and pose changes: jaw close/open, yaw rotation, smile, pucker and eyebrow lower/raise.

We also show examples of expression and pose modification on faces from existing people in more challenging conditions with variable lighting conditions, different camera and head poses and different image quality in Figure 6. The people in these samples have explicitly granted consent for their use in this work and for their expressions and poses to be modified.

Finally, we show an attached video of results generated by our network. Note that the view correspondence is good, even when no temporal information is used by the network.

A.4. Comparison with Zakharov et al. [7]

All the samples generated using Zakharav et al. [7] use their generator in one-shot mode for a direct comparison with our method. It should be noted that most of the visual results in Zakharav et al. [7] are generated using few-shot mode, where multiple samples of the source

identity are available, which leads to a better quality of generated images. Due to lack of official source code, we use an open-source version of the code [1] that has been trained for five epochs in a smaller version of the full VoxCeleb2 [2].

[1] github.com/vincent-thevenin/Realistic-Neural-Talking-Head-Models

Figure 1: Comparison between (1) our face synthesis network MorphGAN, (2) an unofficial open-source implementation of Zakharov et al. [7] and (3) the official implementation of Wiles et al. [6]. The presented expression and pose changes are for "jaw close/open" sequence. For each example, The pink highlighted image is the reference image.

Figure 2: Comparison between (1) our face synthesis network MorphGAN, (2) an unofficial open-source implementation of Zakharov et al. [7] and (3) the official implementation of Wiles et al. [6]. The presented expression and pose changes are for "yaw rotation" sequence. For each example, The pink highlighted image is the reference image.

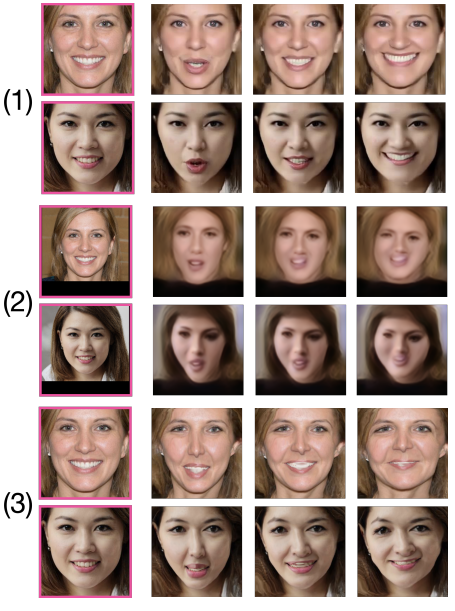

Figure 3: Comparison between (1) our face synthesis network MorphGAN, (2) an unofficial open-source implementation of Zakharov et al. [7] and (3) the official implementation of Wiles et al. [6]. The presented expression and pose changes are for "smile" sequence. For each example, the pink highlighted image is the reference image.

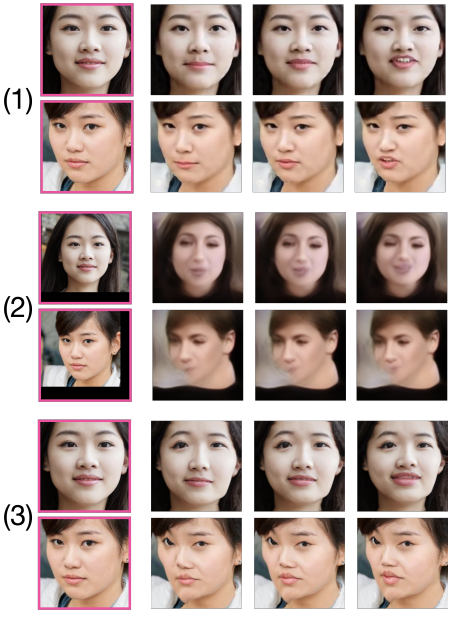

Figure 4: Comparison between (1) our face synthesis network MorphGAN, (2) an unofficial open-source implementation of Zakharov et al. [7] and (3) the official implementation of Wiles et al. [6]. The presented expression and pose changes are for "pucker" sequence. For each example, the pink highlighted image is the reference image.

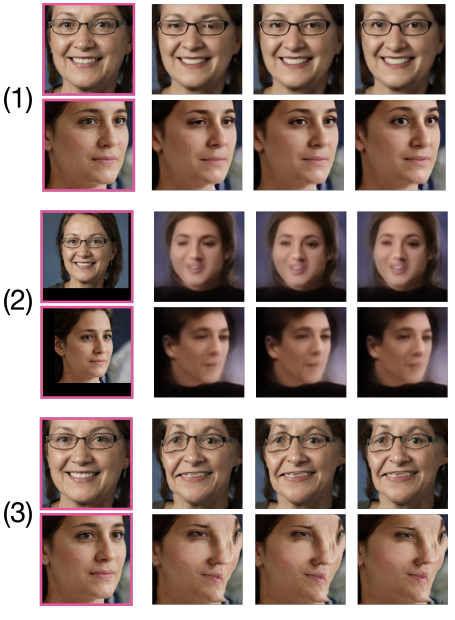

Figure 5: Comparison between (1) our face synthesis network MorphGAN, (2) an unofficial open-source implementation of Zakharov et al. [7] and (3) the official implementation of Wiles et al. [6]. The presented expression and pose changes are for "eyebrow lower/raise" sequence. For each example, the pink highlighted image is the reference image.

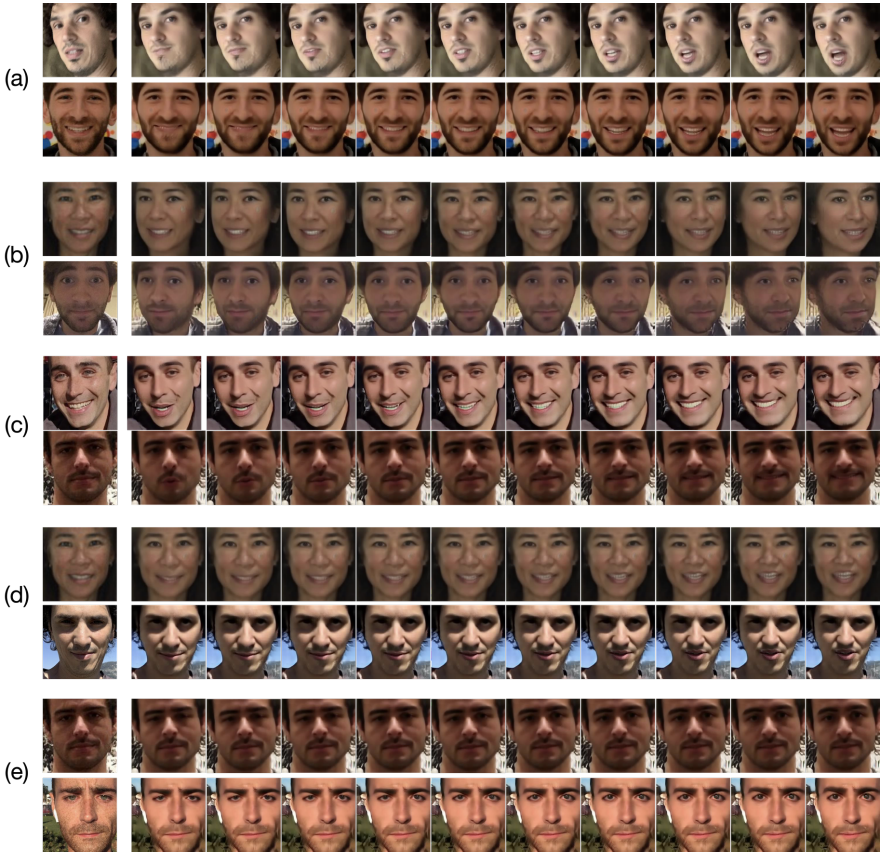

Figure 6: Examples of facial expression and head pose changes using MorphNet on example faces of real people that explicitly granted permission for use as shown in this work. The presented expression and pose changes are *(a)* jaw close/open, *(b)* yaw rotation, *(c)* smile, *(d)* pucker and *(e)* eyebrow lower/raise.

References

[1] Y. Choi, Y. Uh, J. Yoo, and J. Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020.

[2] J. Chung, A. Nagrani, and A. Zisserman. VoxCeleb2: Deep speaker recognition. In *Interspeech*, 2018.

[3] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1501–1510, 2017.

[4] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[6] O. Wiles, S. Koepke, and A. Zisserman. X2Face: A network for controlling face generation using images, audio, and pose codes. In *Proceedings of the European Conference on Computer Vision*, pages 670–686, 2018.

[7] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky. Few-shot adversarial learning of realistic neural talking head models. *arXiv preprint arXiv:1905.08233*, 2019.