Supplementary Material for Cascade EF-GAN

Rongliang Wu[1], Gongjie Zhang[1], Shijian Lu[1], and Tao Chen[2]

[1]Nanyang Technological University
[2]Fudan University
ronglian001@e.ntu.edu.sg, {gongjiezhang,shijian.lu}@ntu.edu.sg, eetchen@fudan.edu.cn

1. Loss Function

The loss function for training the proposed EF-GAN contains five terms: 1) the adversarial loss for improving the photo-realism of the synthesized facial expression images to make them indistinguishable from real samples; 2) the conditional expression loss to ensure generated facial expression images to align with the provided target AUs labels; 3) the content loss for preserving the identity information and consistency of the image content. 4) the attention loss to encourage the attentive module to produce sparse attention map and pay attention to the regions that really need modification. 5) the interpolation loss to constrain the interpolated AUs label has desired sematical meaning and resides on the manifold of natural AUs.

Formally, given a facial expression image and its corresponding local regions $I_x = \{I_{face}, I_{eyes}, I_{nose}, I_{mouth}\}$ with AUs label y_x and the expression residual r. The target expression AUs label y_z is generated by the Interpolator $y_z = Interp(y_x, r)$. The discriminator D_{interp} is trained to distinguish real/fake AUs. The initial output produced by the Expression Transformer is $I_z^{init} = ET(I_x, y_z)$, where $I_z^{init} = \{I_{face}^{init}, I_{eyes}^{init}, I_{nose}^{init}, I_{mouth}^{init}\}$ and $ET = \{ET_{face}, ET_{eye}, ET_{nose}, ET_{mouth}\}$. We then feed the initial outputs to the refiner, and the final output is generated by $I_z = R(I_z^{init})$. We define $I = \{I_z, I_z^{init}\}$ to simplify the expression in the following section. The discriminator D distinguishes whether the query image is a real image or not. To improve the quality of the synthesized image, We introduce a hierarchical $D = \{D_{final}, D_{init}\}$, where $D_{init} = \{D_{face}, D_{eye}, D_{nose}, D_{mouth}\}$ is a set of four discriminators for the initial outputs. D_{final} examines the final output to judge the holistic of facial features and predict the AUs label, while D_{init} examine the quality of initial outputs.

Adversarial Loss We adopt the WGAN-GP [1] to learn the parameters. The adversarial loss function is formulated as:

$$\begin{aligned}\mathcal{L}_{adv} = \sum_i \{&\mathbb{E}_{I_{x_i}\sim P_{data}}[\log D_i(I_{x_i})] - \mathbb{E}_{I_i\sim P_S}[\log D_i(I_i)] \\ &- \lambda_{gp}\mathbb{E}_{\tilde{I}_i\sim P_{\tilde{I}_i}}[\|\nabla_{\tilde{I}_i} D_i(\tilde{I}_i)\|_2 - 1]^2\},\end{aligned} \tag{1}$$

where $D_i \in D$, $I_{x_i} \in I_x$, $I_i \in I$, P_{data} stands for the data distribution of the real images, P_S the distribution of the synthesized images and $P_{\tilde{I}_i}$ the random interpolation distribution. λ_{gp} is set to be 10.

Conditional Expression Loss For a given input I_x and the target expression label y_z, our goal is to synthesize an output image I_z with the desired target expression. To achieve this condition, we add an auxiliary classifier on top of D and impose AUs regression loss when training the network. In particular, the objective is composed of two terms: an AUs regression loss with generated images used to optimize the parameters of Expression Transformer and Refiner, and an AUs regression loss of real images used to optimize Discriminator D_{final}. In detail, the loss is formulated as:

$$\begin{aligned}\mathcal{L}_{cond} =&\mathbb{E}_{I_x\sim P_{data}}[\|D_{final}(I_x) - y_x\|_2^2] \\ &+ \mathbb{E}_{I_z\sim P_S}[\|D_{final}(I_z) - y_z\|_2^2].\end{aligned} \tag{2}$$

Content Loss In order to guarantee that the face in both the input and output images correspond to the same person, we adopt cycle loss [5] to force the model to maintain the identity information and personal content after the expression editing process by minimizing the $L1$ difference between the original image and its reconstruction:

$$\mathcal{L}_{cont} = \mathbb{E}_{I_x\sim P_{data}}[\|I_{rec} - I_x\|_1]. \tag{3}$$

Note that the content loss is only applied to original input and final output image.

Attention Loss To encourage the attentive module to produce sparse attention map and pay attention to the regions that really need modification rather than the whole image,

we introduce a sparse loss over the attention map:

$$\mathcal{L}_{attn} = \mathbb{E}_{x \in X}[\|M_A(I_{face})\|_2 + \|M_A(I_{eye})\|_2 + \|M_A(I_{nose})\|_2 + \|M_A(I_{mouth})\|_2]. \quad (4)$$

Interpolation Loss The interpolation loss is consist of two terms: the regression term to make it has the similar semantic meaning with the pseudo AUs target y_p (generated by linear interpolation), and the regularized term to constrain it reside on the manifold of natural AUs:

$$\mathcal{L}_{interp} = \mathbb{E}_{\hat{y} \sim P_I}[\|\hat{y} - y_p\|_2 + \lambda_{int}\mathbb{E}_{\hat{y} \sim P_R}[logD_{interp}(\hat{y})], \quad (5)$$

where $\hat{y}$ stands for the interpolated AUs, P_I the data distribution of the interpolated AUs and P_R the distribution of the real AUs. D_{interp} is the discriminator for AUs, which is also trained with WGAN-AP. λ_{int} is set to be 0.1.

Overall Objective Function Finally, the overall objective function is expressed as:

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda_1\mathcal{L}_{cond} + \lambda_2\mathcal{L}_{cont} + \lambda_3\mathcal{L}_{attn} + \lambda_4\mathcal{L}_{interp} \quad (6)$$

where λ_1, λ_2, λ_3 and λ_4 are the hyper-parameters that control the relative importance of every loss term.

In Cascade EF-GAN, the total loss is the sum of the loss of each EF-GAN with equal weights.

2. Network Architecture

In EF-GAN, We have one global branch that captures the global facial structures and three local focus branches that help better preserve identity-related features as well as local details around eyes, noses and mouths. Each branch shares similar architecture without sharing weights. The detailed network architecture is illustrated in Fig. 1. Note that all the convolutional layers in Expression Transformer and Refiner are followed by Instance Normalization layer and ReLU activation layer (omitted in the figure for simplicity), except for the output layer. And the convolutional layers of Discriminator are followed by Leakly ReLU activation layer with slope of 0.01. The number of bottleneck layers of the discriminator for global face images is set to 5 while that of local regions is set to 3 according to the size of the images. The AUs prediction layer is only applied to the final output.

We stack multiple EF-GAN modules in sequence to form Cascade EF-GAN.

3. Training Details

EF-GAN Training Details. We adopt Adam optimizer [2] with β_1 = 0.5, β_2 = 0.999 for EF-GAN optimization. We set λ_1, λ_2 λ_3, λ_4 to be 3000, 10, 0.1 and 1 to balance the magnitude of the losses. The batchsize is set to 2. The total number of epochs is set to 100. The initial learning rate is set to 1e-4 for the first 50 epochs, and linearly decay to 0 over the remaining epochs. Beyond that, we apply Orthogonal Initialization [4] and Spectral Normalization [3] in all convolutional layers except the output layer to stabilize the training process.

Cascade EF-GAN Training Details. To train the Cascade EF-GAN, we first use the weights of a well-trained EF-GAN model to initialize each EF-GAN module in the cascade. Then we train the Cascade EF-GAN model end-to-end for the first 10 epochs, with learning rate starting from 1e-5 and linearly decayed to 0 over the remaining epochs.

Training Time. We use a Tesla V100 GPU in training. For RaFD dataset, it takes 13 hours in training EF-GAN and 8 hours for fine-tuning the Cascade EF-GAN structure. For CFEED dataset, it takes 33 and 20 hours in training and fine-tuning, respectively.

4. More Results

We have also presented more results generated by our proposed Cascade EF-GAN in the following pages.

References

[1] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[3] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[4] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[5] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

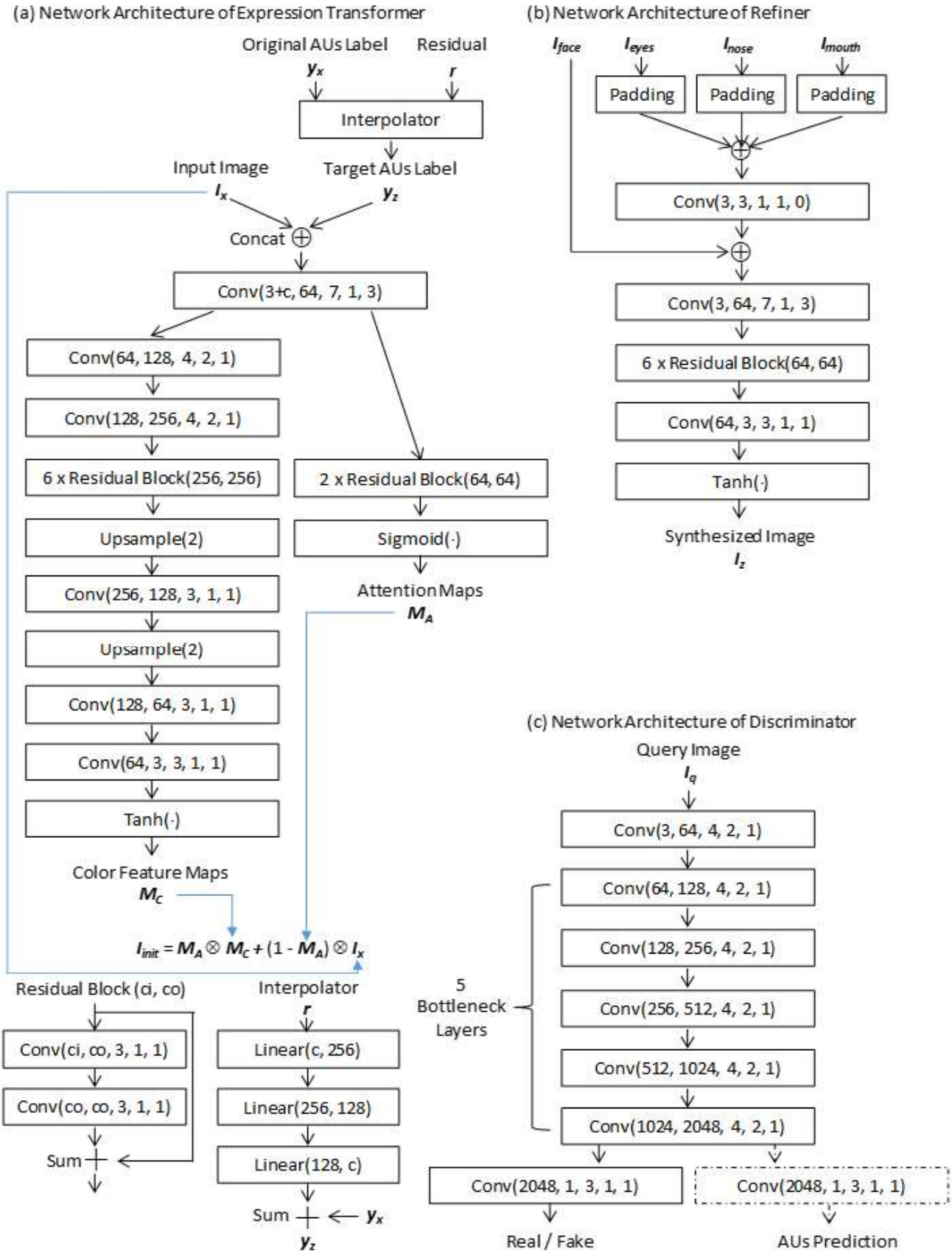

Figure 1. The detailed network architecture of our proposed model. (a-c) shows the architecture of Expression Transformer, Refiner and Discriminator, respectively. Conv(Nin, Nout, k, s, p) denotes a convolutional layer whose input channel number is Nin, output channel number is Nout, kernel size is k, stride is y and padding is p. Linear(Nin, Nout) denotes a fully connected layer with Nin and Nout as its input and output channel number, respectively. Parameter c denotes dimension of AUs label.

Figure 2. Additional expression editing results on wild images. In each triplet, the first column is input facial image, the second column is the image with desired expression and the last column is the synthesized result.

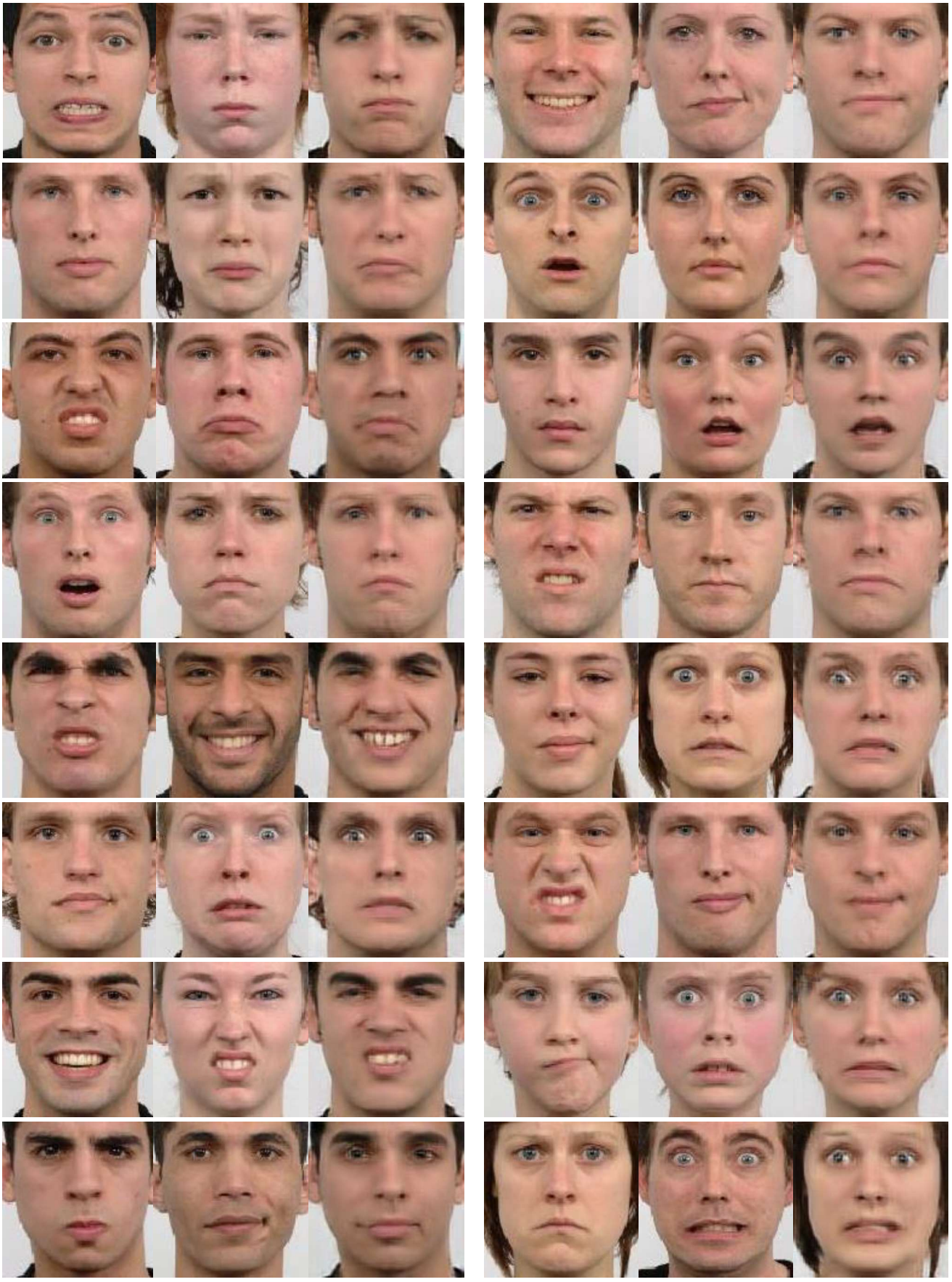

Figure 3. Additional expression editing results on RaFD. In each triplet, the first column is input facial image, the second column is the image with desired expression and the last column is the synthesized result.

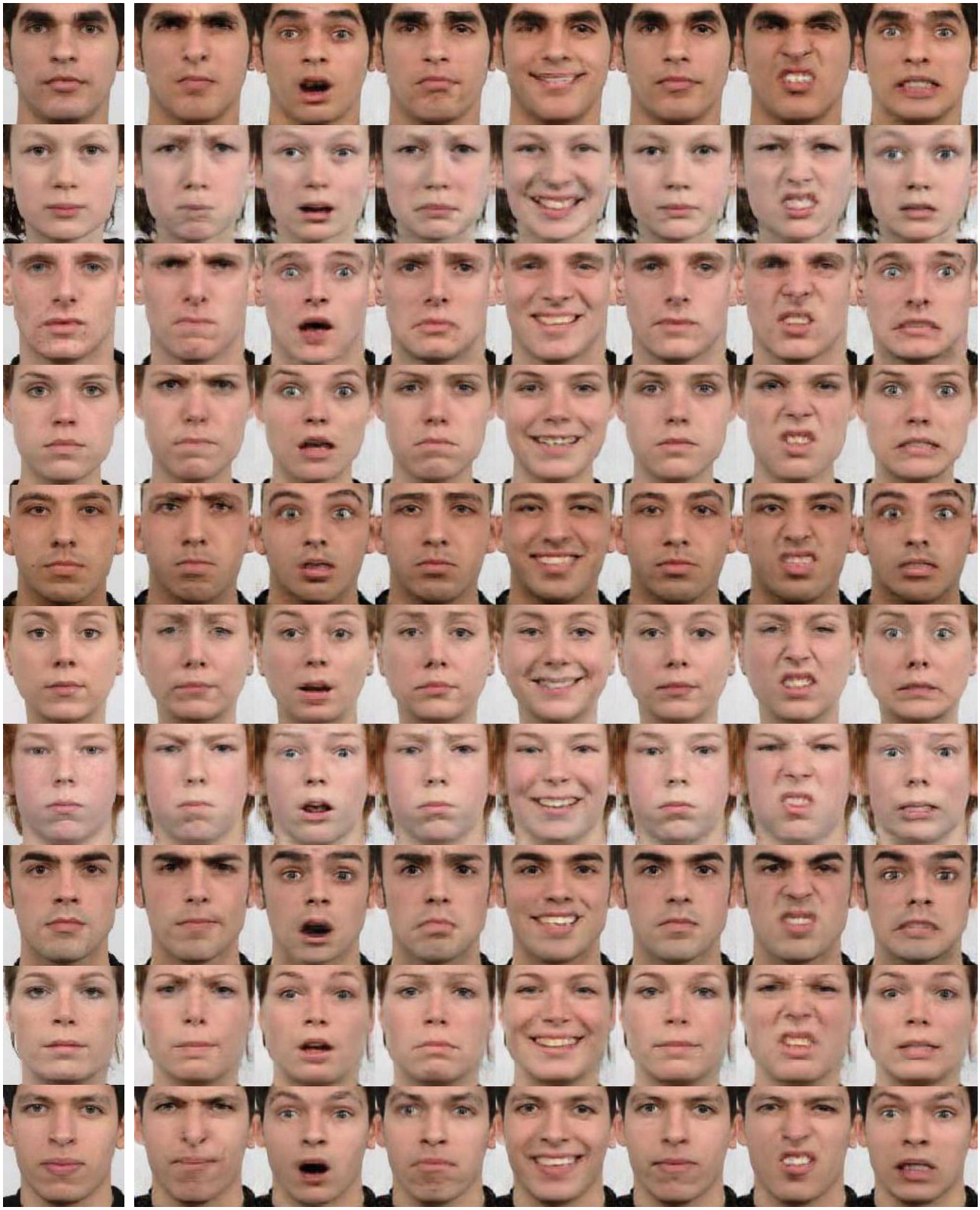

Figure 4. Additional expression editing results on RaFD (Input, Angry, Surprised, Sad, Happy, Neutral, Disgusted, Fearful).

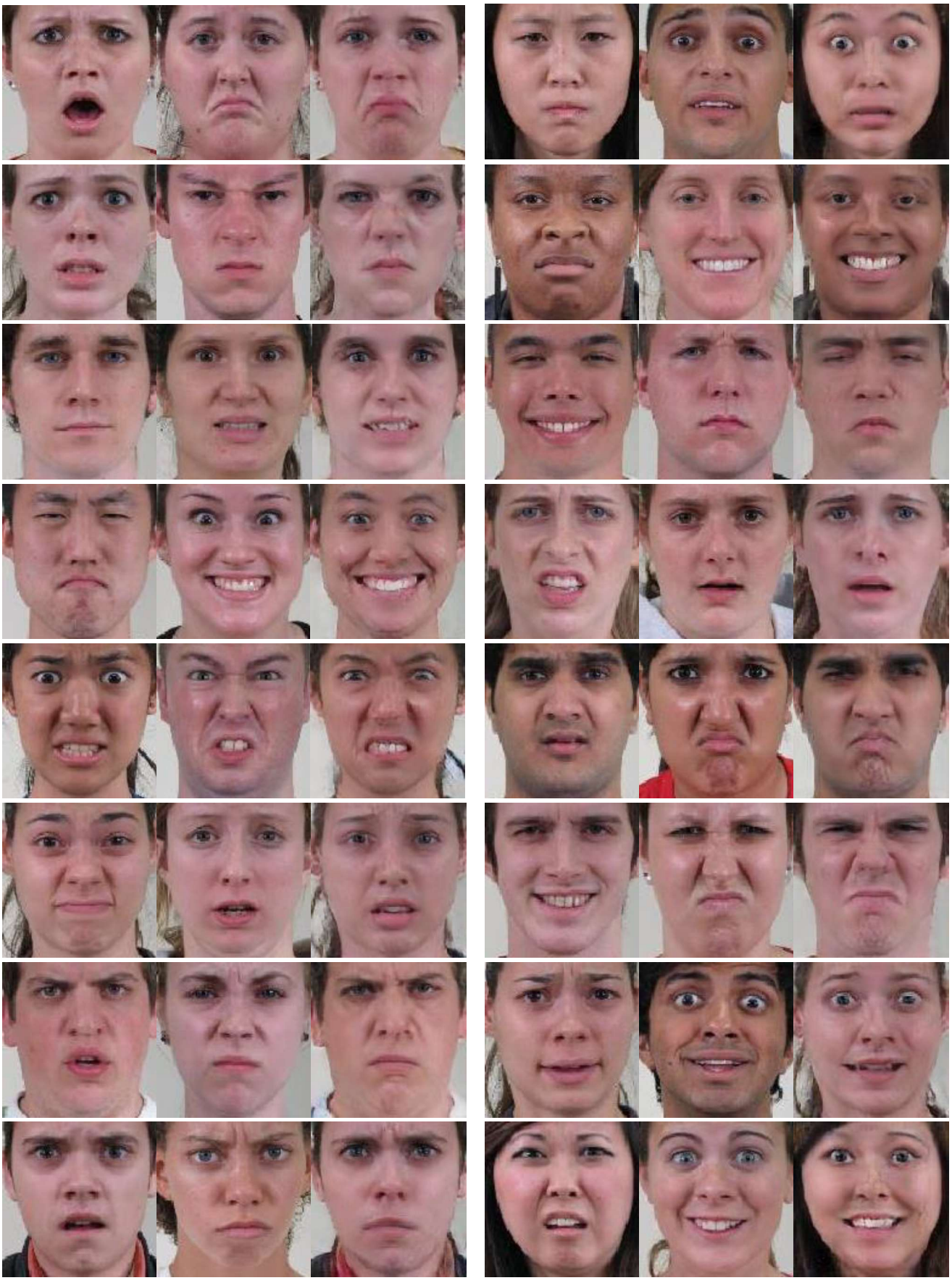

Figure 5. Additional expression editing results on CFEED. In each triplet, the first column is input facial image, the second column is the image with desired expression and the last column is the synthesized result.

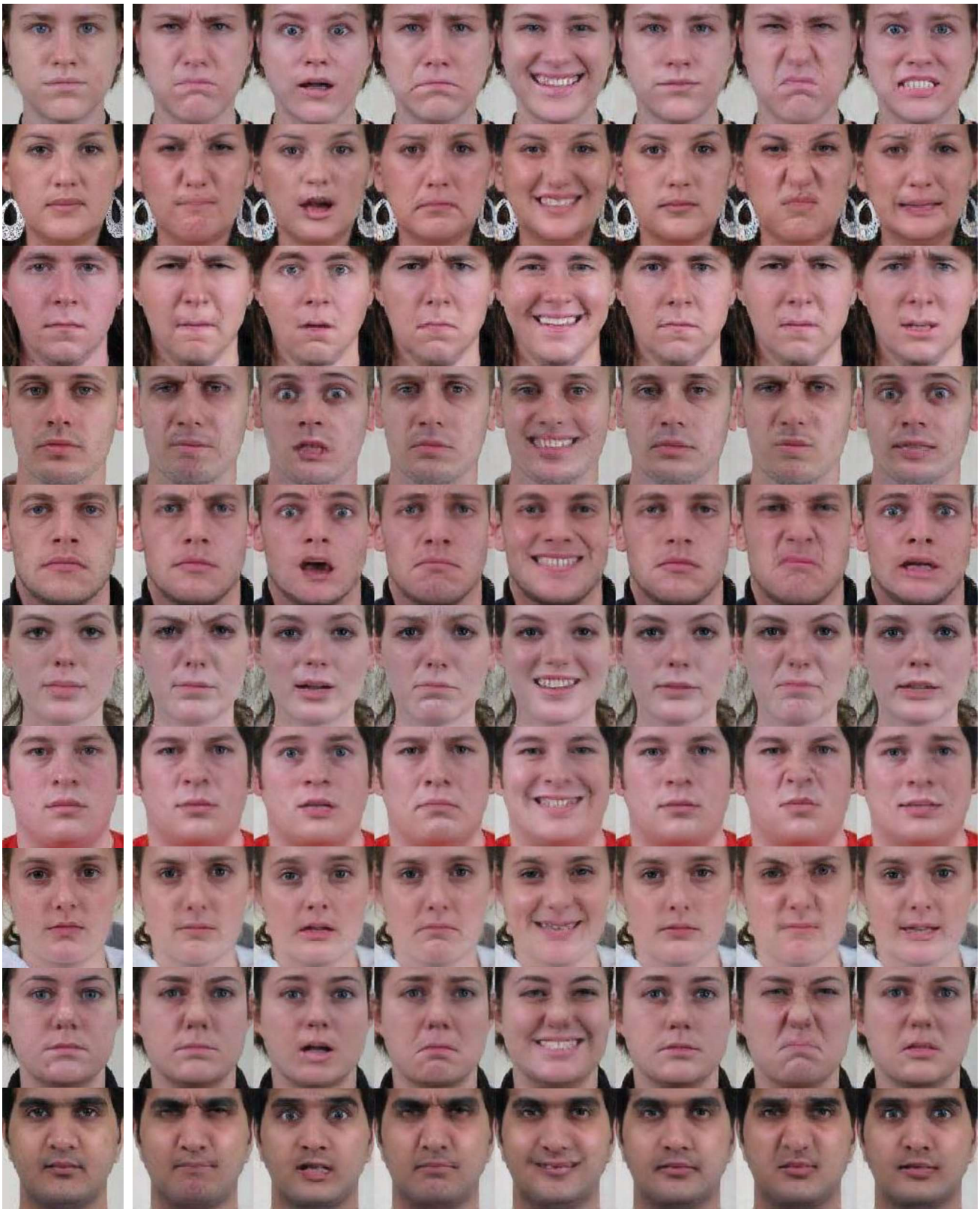

Figure 6. Additional expression editing results on CFEED (Input, Angry, Surprised, Sad, Happy, Neutral, Disgusted, Fearful).