Supplementary materials: WP2-GAN: Wavelet-based Multi-level GAN for Progressive Facial Expression Translation with Parallel Generators

Jun Shao
sh_jun@encs.concordia.ca
Tien D. Bui
bui@cse.concordia.ca

Computer Science and Software Engineering
Concordia University
Montréal, Québec, Canada

1 Structures of Neural Networks

	Layer	Output Size	Details
Input	ConvI	$128 \times 128 \times 64$	K7, S1, P3
Down-sampling	ConVD1	$64 \times 64 \times 128$	K4, S2, P1
	ConVD2	$32 \times 32 \times 256$	K4, S2, P1
	ConVD3	$16 \times 16 \times 512$	K4, S2, P1
Residual Blocks	ConvR1-1	$16 \times 16 \times 512$	K3, S1, P1
	ConvR1-2	$16 \times 16 \times 512$	K3, S1, P1
	...	...	...
	ConvR6-1	$16 \times 16 \times 512$	K3, S1, P1
	ConvR6-2	$16 \times 16 \times 512$	K3, S1, P1
Up-sampling	ConvU1	$32 \times 32 \times 256$	K4, S2, P1
	ConvU2	$64 \times 64 \times 128$	K4, S2, P1
	ConvU3	$128 \times 128 \times 64$	K4, S2, P1
Output	ConVO1	$128 \times 128 \times 3$	K7, S1, P3
	ConVO2	$128 \times 128 \times 1$	K7, S1, P3

Table 1: Neural network architecture of generator (G_A/G_B) ('K','S' and 'P' denote 'Kernel size','stride' and 'padding' of convolutional layers).

The structure of the generator and three discriminators are shown in Table 1 and Table 2. It is worth noting that there are InstanceNorm and ReLU layers between the convolutional layers in the generator. A Tanh layer is posed on the outcome of ConVO1 to generate the color map, while a Sigmoid layer is leveraged to produce mask map from the outcome of ConVO2. Additionally, there are InstanceNorm and LeakyReLU layers between the convolutional layers in the discriminators.

	Layer	Output Size			Details
		D1	D2	D3	
Input	ConvI	$64 \times 64 \times 64$	$32 \times 32 \times 64$	$16 \times 16 \times 128$	K4, S2, P1
Down-sampling	ConVD1	$32 \times 32 \times 128$	$16 \times 16 \times 128$	$8 \times 8 \times 128$	K4, S2, P1
	ConVD2	$16 \times 16 \times 256$	$8 \times 8 \times 256$	$4 \times 4 \times 256$	K4, S2, P1
	ConVD3	$8 \times 8 \times 512$	$4 \times 4 \times 512$	$2 \times 2 \times 512$	K4, S2, P1
	ConVD4	$4 \times 4 \times 1024$	$2 \times 2 \times 1024$	—	K4, S2, P1
	ConVD5	$2 \times 2 \times 2048$	—	—	K4, S2, P1
Output	ConvO-Adv	$2 \times 2 \times 1$	$2 \times 2 \times 1$	$2 \times 2 \times 1$	K3, S1, P1
	ConvO-Aus	$1 \times 1 \times 17$	$1 \times 1 \times 17$	$1 \times 1 \times 17$	K2, S1, P0

Table 2: Neural network architecture of multi-level discriminators ('K','S' and 'P' denote 'Kernel size', 'stride' and 'padding' of convolutional layers).

2 Loss Functions

To make the generated images indistinguishable from real images, we adopt the adversarial loss proposed by WGAN-GP [7] for each step of progressive training, which is defined as:

$$\mathcal{L}_{avd} = \frac{1}{3}\sum_{i=1}^{3}\left\{\frac{1}{2}\mathbb{E}_{\mathbf{x}_{in}\sim\bar{\mathbb{P}}}\left[D_i\left(G_A\right)+D_i\left(G_B\right)\right]-\mathbb{E}_{\mathbf{x}_o\sim\mathbb{P}}\left[D_i\left(\mathbf{x}_o\right)\right]+\lambda_{gp}\mathbb{E}_{\widetilde{\mathbf{x}}\sim\widetilde{\mathbb{P}}}\left(\parallel\nabla_{\widetilde{\mathbf{x}}}D_i\left(\widetilde{\mathbf{x}}\right)\parallel_2-1\right)^2\right\}, \tag{1}$$

where $G_m = G_m\left(\mathbf{x}_{in}|\mathbf{y}_t\right), m \in \{A,B\}$. $\mathbb{P}$ is the data distribution of original image $\mathbf{x}_0$, $\widetilde{\mathbf{x}}$ represents the random interpolated image by input image $\mathbf{x}_0$ and the generated face $\mathbf{x}_t$, $\widetilde{\mathbb{P}}$ stands for the uniform interpolation distribution, $\bar{\mathbb{P}}$ indicates the union of $\mathbb{P}$ and $\widetilde{\mathbb{P}}$, while λ_{gp} is a penalty coefficient.

According to [5], the attention mask A is easy to saturate to 1, making $G\left(\mathbf{x}_{in}|\mathbf{y}_t\right) = \mathbf{x}_{in}$. To prevent the saturation, l_2-weight penalty is added to the attention mask. A Total Variation Regularization is imposed on A to enforce the generated images to be smooth. The attention loss can be defined as:

$$\mathcal{L}_A(G,\mathbf{x}_{in},\mathbf{y}_t) = \frac{1}{2}\sum_{M\in\{M_A,M_B\}}\left\{\mathbb{E}_{\mathbf{x}_{in}\sim\bar{\mathbb{P}}}\left[\parallel M\parallel_2\right]+\lambda_{TV}\mathbb{E}_{\mathbf{x}_{in}\sim\bar{\mathbb{P}}}\left[\sum_{i,j}^{H,W}\left(M_{i+1,j}-M_{i,j}\right)^2+\left(M_{i,j+1}-M_{i,j}\right)^2\right]\right\}, \tag{2}$$

where M_A and M_B are masks generated by generator G_A and G_B, respectively. λ_{TV} is the penalty coefficient for mask smoothing.

Besides the adversarial loss and the attention loss, the generator and the discriminator also have to reduce the errors produced by the regression layer imposed on top of each critic. The whole condition losses can be written as:

$$\mathcal{L}_{cond} = \frac{1}{3}\sum_{i=1}^{3}\left\{\frac{1}{2}\sum_{m\in\{A,B\}}\mathbb{E}_{\mathbf{x}_{in}\sim\bar{\mathbb{P}}}\left[\parallel D_i(G_m)-\mathbf{y}_t\parallel_2^2\right]+\mathbb{E}_{\mathbf{x}_o\sim\mathbb{P}}\left[\parallel D_i(\mathbf{x}_o)-\mathbf{y}_o\parallel_2^2\right]\right\}, \tag{3}$$

where $G_m = G_m\left(\mathbf{x}_{in}|\mathbf{y}_t\right), m \in \{A,B\}$ and *cond* is AUs code. $\mathbf{y}_t$ and $\mathbf{y}_{in}$ are target and input condition (expression).

To ensure two generators G_A and G_B proceed in the same direction, we impose a similarity loss to the forward outcomes of two generators. The similarity loss is formulated as:

$$\mathcal{L}_{sim} = \mathbb{E}_{\mathbf{x}_{in}\sim\bar{\mathbb{P}}}\left[\parallel G_A(\mathbf{x}_{in}|\mathbf{y}_t)-G_B(\mathbf{x}_{in}|\mathbf{y}_t)\parallel_1\right], \tag{4}$$

Minimizing the loss functions above does not guarantee that the generated images keep the same identity with their input counterparts. A cycle-consistent loss [6] is utilized by the generators to preserve the identity-level consistency. We adopt the l_1 norm, which helps to capture features associated with low-frequencies. The cycle-consistent loss is formulated as:

$$\mathcal{L}_{cyc} = \frac{1}{2}\mathbb{E}_{\mathbf{x}_{in}\sim\mathbb{P}}\Big[\parallel G_B(G_A(\mathbf{x}_{in}|\mathbf{y}_t)|\mathbf{y}_{in}) - \mathbf{x}_{in} \parallel_1 + \parallel G_A(G_B(\mathbf{x}_{in}|\mathbf{y}_t)|\mathbf{y}_{in}) - \mathbf{x}_{in} \parallel_1 \Big], \quad (5)$$

The overall loss functions for G and D can be formulated as:

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda_{cond}\mathcal{L}_{cond} + \lambda_{sim}\mathcal{L}_{sim} + \lambda_{cyc}\mathcal{L}_{cyc} + \lambda_A(\mathcal{L}_A(G, \mathbf{x}_{in}, \mathbf{y}_t) + \mathcal{L}_A(G, \mathbf{x}_t, \mathbf{y}_{in})), \quad (6)$$

where λ_{cond}, λ_{sim}, λ_{cyc} and λ_A are hyper-parameters that control the relative importance of conditional loss, similarity loss, cycle-consistent loss and attention loss, respectively.

3 Implementation Details

Our approach adopts three steps of progressive training. We utilize Adam [3] for the model optimization with the following hyper-parameters: learning rate=0.00005, beta1=0.5, beta2 =0.999 and batch size=25. During progressive training, the generators and discriminators are optimized at the same frequency. The weight coefficients for the loss functions are set to $\lambda_A = 0.2$, $\lambda_{gp} = 10$, $\lambda_{cond} = 160$, $\lambda_{sim} = 1$ and $\lambda_{cyc} = 10$.

Our model is trained on RafD [4] and CFEED [1] for 200 epochs, respectively. The learning rate (lr) is linearly decayed to zero over the last 50 epochs of the training.

For the parallel only training experiments, the generator is optimized once after five times optimization of the discriminators, with an initial learning rate of 0.0001. The weight coefficient that is different from the progressive training is $\lambda_{sim} = 5$.

Our progressive training model is trained on RafD and CFEED for about 13h and 40h, respectively, with a single Tesla V100 GPU.

4 More Experimental Results

We extensively test the impact of using immediate result instead of original input as resource to compute background information of the translation. Results show that this practice leads to the drop of both translation accuracy (84.30%) and image quality (FID: 31.02) on CFEED.

We further explore the impact of steps in progressive training and train a model on CFEED adopting four steps. The results show that further increase of progressive training steps fails to bring an improvement to both translation accuracy and image quality (Accuracy: 87.23%, FID: 24.62, SSIM: 0.6753). Thus, we adopt three steps for our proposed method.

More results of expression translation by our proposed model on EmotioNet, RafD and CFEED are shown in Figure 1$\sim$ Figure 3.

References

[1] Shichuan Du, Yong Tao, and Aleix M. Martinez. Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 111(15):E1454–E1462, Apr.

2014. ISSN 0027-8424. doi: 10.1073/pnas.1322355111. URL http://www.pnas.org/cgi/doi/10.1073/pnas.1322355111.

[2] 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.

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

[4] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel H. J. Wigboldus, Skyler T. Hawk, and Ad van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 24(8):1377–1388, Dec. 2010. ISSN 0269-9931. doi: 10.1080/02699930903485076. URL http://www.tandfonline.com/doi/abs/10.1080/02699930903485076.

[5] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: One-shot anatomically consistent facial animation. *International Journal of Computer Vision*, pages 1–16, 2019.

[6] 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: Supplementary expression translation results by our proposed model on EmotioNet. In each triplet, the first column is the test face, followed by an image with the target expression and finally the synthesized image.

Figure 2: Supplementary results of expression translation by our proposed model on RafD (left) and CFEED (right). In each triplet, the first column is the test face, followed by an image with the target expression and finally the synthesized image.

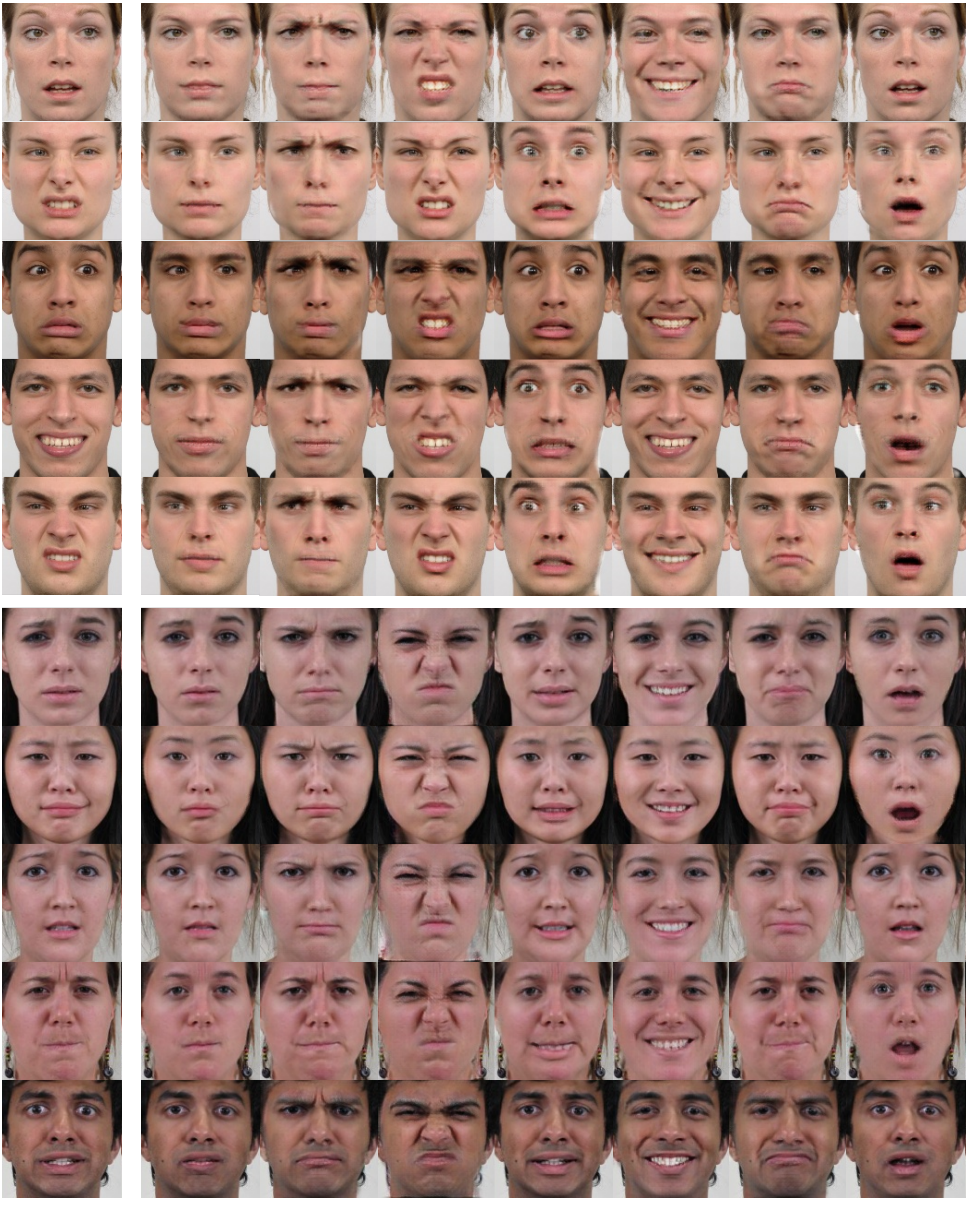

Figure 3: Supplementary results of seven basic expressions synthesized by our proposed model (Input, Neutral, Angry, Disgusted, Fearful, Happy, Sad, Surprised) on RafD (top) and CFEED (bottom).