Supplementary Material: Geometry Driven Progressive Warping for One-Shot Face Animation

Yatao Zhong
yazhong@microsoft.com
Faezeh Amjadi
faamja@microsoft.com
Ilya Zharkov
zharkov@microsoft.com

Applied Sciences, Microsoft
Redmond, WA, USA

1 Training Losses

In this section we provide the details of the loss functions that are used during training. We denote the ground truth driving image by I_d and generated output image by I_o.

1.1 Perceptual loss

We adopt a VGG-19 [8] based perceptual loss [6]. A pyramid of three scales (256x256, 128x128, 64x64) of input images are used and the loss is calculated on the convolutional outputs (instead of the pooling outputs) of the first five blocks.

$$\mathcal{L}_{\mathcal{P}}(I_d, I_o) = \sum_s \sum_j \sum_i |\mathcal{V}_{j,i}(I_d^s) - \mathcal{V}_{j,i}(I_o^s)|, \tag{1}$$

where s denotes the image scale and $\mathcal{V}_{j,i}(\cdot)$ is the j-th layer output of VGG-19 at spatial location i.

1.2 Adversarial loss

To generate photo realistic images, we train the model with Patch-GAN [5] and Hinge loss [7]. Similar to perceptual loss, we apply adversarial training on a pyramid of three scales (256x256, 128x128, 64x64) of input images.

$$\mathcal{L}_{\mathcal{G}} = -\sum_s \sum_i \mathcal{D}_i(I_o^s), \tag{2}$$

$$\mathcal{L}_{\mathcal{D}} = -\sum_s \sum_i \left[min(0, -1 - \mathcal{D}_i(I_o^s)) + min(0, -1 + \mathcal{D}_i(I_d^s))\right], \tag{3}$$

© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

where s denotes the image scale, i is the spatial location of Patch-GAN discriminator output, $\mathcal{L}_{\mathcal{G}}$ and $\mathcal{L}_{\mathcal{D}}$ are the generator loss and discriminator loss respectively.

1.3 Feature matching loss

To stabilize adversarial training, we also adopt discriminator feature matching loss between the real images and generated images. Again, the loss is calculated on a pyramid of three image scales.

$$\mathcal{L}_{\mathcal{M}}(I_d, I_o) = \sum_s \sum_j \sum_i |\mathcal{D}_{j,i}(I_d^s) - \mathcal{D}_{j,i}(I_o^s)|, \tag{4}$$

where s denotes the image scale, $\mathcal{D}_{j,i}(\cdot)$ denotes the j-th layer output of the discriminator $\mathcal{D}$ at spatial location i.

1.4 Additional warping constraints

So far we have described how we compute the losses for the generated output image I_o. To add more constraints for the model to learn correct displacements, we take the predicted displacement field from the last pyramid in the decoder and use that to warp the source image to create a warped image I_w. This image is less realistic than I_o, but still can provide some additional supervision to train the model. Therefore, we take the warped image I_w and compute all of the above losses between I_w and I_d.

2 Implementation Details

Here we show the implementation details of the proposed model in Fig. 1. The model adopts an encoder-decoder structure. For the model variant *"PWM + geom. disp."*, the guidance map shown in the figure would be the rendered geometric displacement field and the input would be the source image only. For the model variant *"PWM + neural codes"*, the posed neural codes (rendered in target image space) would be used as the guidance map. In this case, we also render the latent codes in source image space and concatenate it with the source image before being fed to the encoder.

3 More on Evaluation Metrics

We use the below metrics to evaluate our model and other baselines.

Frechet Inception Distance (FID) [4]. We adopt InceptionV3 [9] as the backbone network and use the output from the last average pooling layer as the input feature vectors to FID. FID measures how close the distribution of generated images is to that of the real images. Therefore it reflects the realism of generated images. For same identity reconstruction, we compute FID between generated images and ground truth target images. For cross identity reconstruction, we compute FID between generated images and input source images. This is to ensure that we measure the realism based on the same identity.

Cosine Similarity (CSIM). We adopt ArcFace [3] as the underlying network to produce the face embedding for comparison. We first detect 68 facial landmarks with [1, 2] and align the face with a predefined reference face to address any ambiguity in scale and rotation.

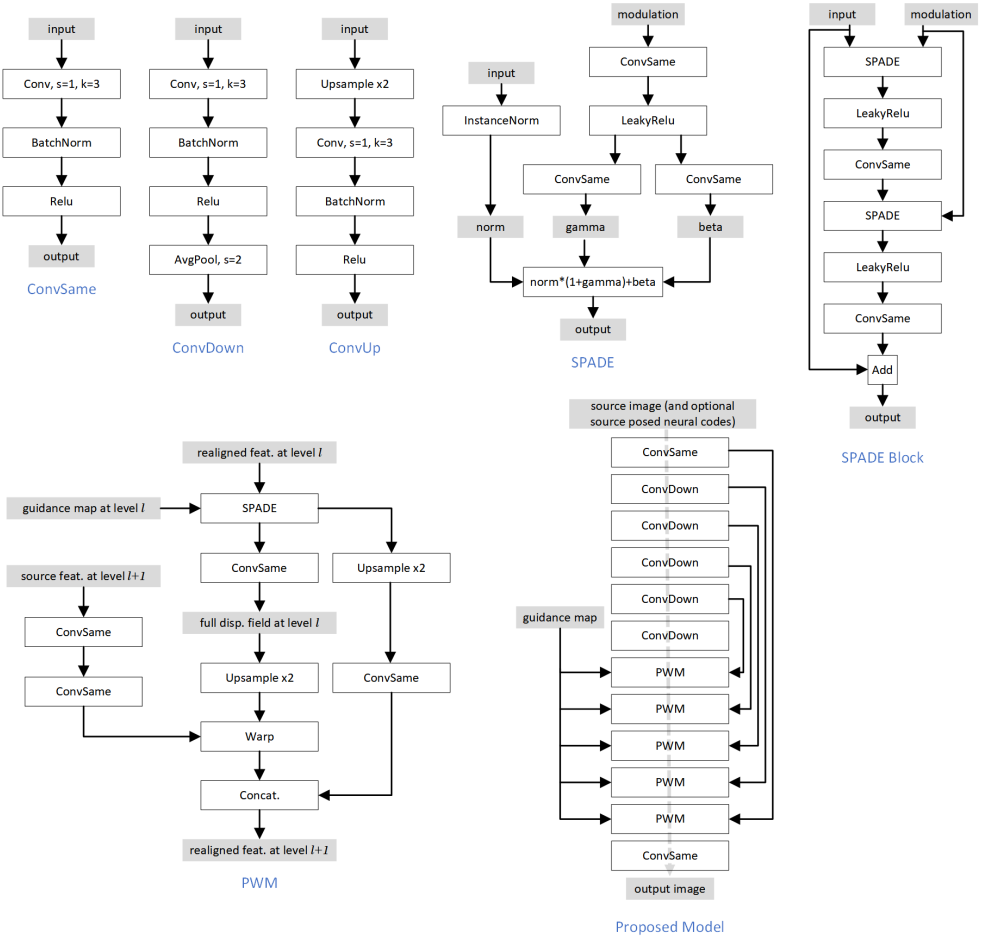

Figure 1: Implementation details of the proposed model.

Then we calculate CSIM on the aligned face images. Because the underlying network is pre-trained for face verification, CSIM can be used to evaluate a model's capability of preserving identity. Similar to FID, we compute CSIM between generated images and ground truth target images for same identity reconstruction and compute CSIM between generated images and input source images for cross identity reconstruction.

Average Keypoint Distance (AKD). We detect 68 facial landmarks [1, 2] and compute the L1 distance between the landmarks of generated images and ground truth target images. AKD is used as a way of measuring the accuracy of motion transfer. It is only used for same identity reconstruction because we do not have the reenacted image of the source identity available as ground truth. Computing AKD between generated images and driving images of different identities is erroneous since facial landmarks encode not only pose information but also person-specific shape information. We should exclude any noise from identities by using the same subject.

Average Pose Distance (APD) and *Average Expression Distance* (AED). We fit FLAME on images and compute the L1 distances of their pose and expression parameters respec-

tively. Note that the pose here refers to the yaw, pitch and roll angles of the global head pose estimated by FLAME. APD and AED are used to evaluate how accurate the model is to transfer the head pose and facial expression. For same identity reconstruction, we compute APD and AED between generated images and the ground truth target images. For cross identity reconstruction, we compute APD and AED between generated images and driving images because they are expected to have the same poses and expressions.

4 Additional Qualitative Results

In this section, we present more qualitative results to show the visual quality of synthesized portrait images. See Fig. 2 for same identity reconstruction and Fig. 3 for cross identity reconstruction.

Figure 2: Qualitative results for same identity reconstruction.

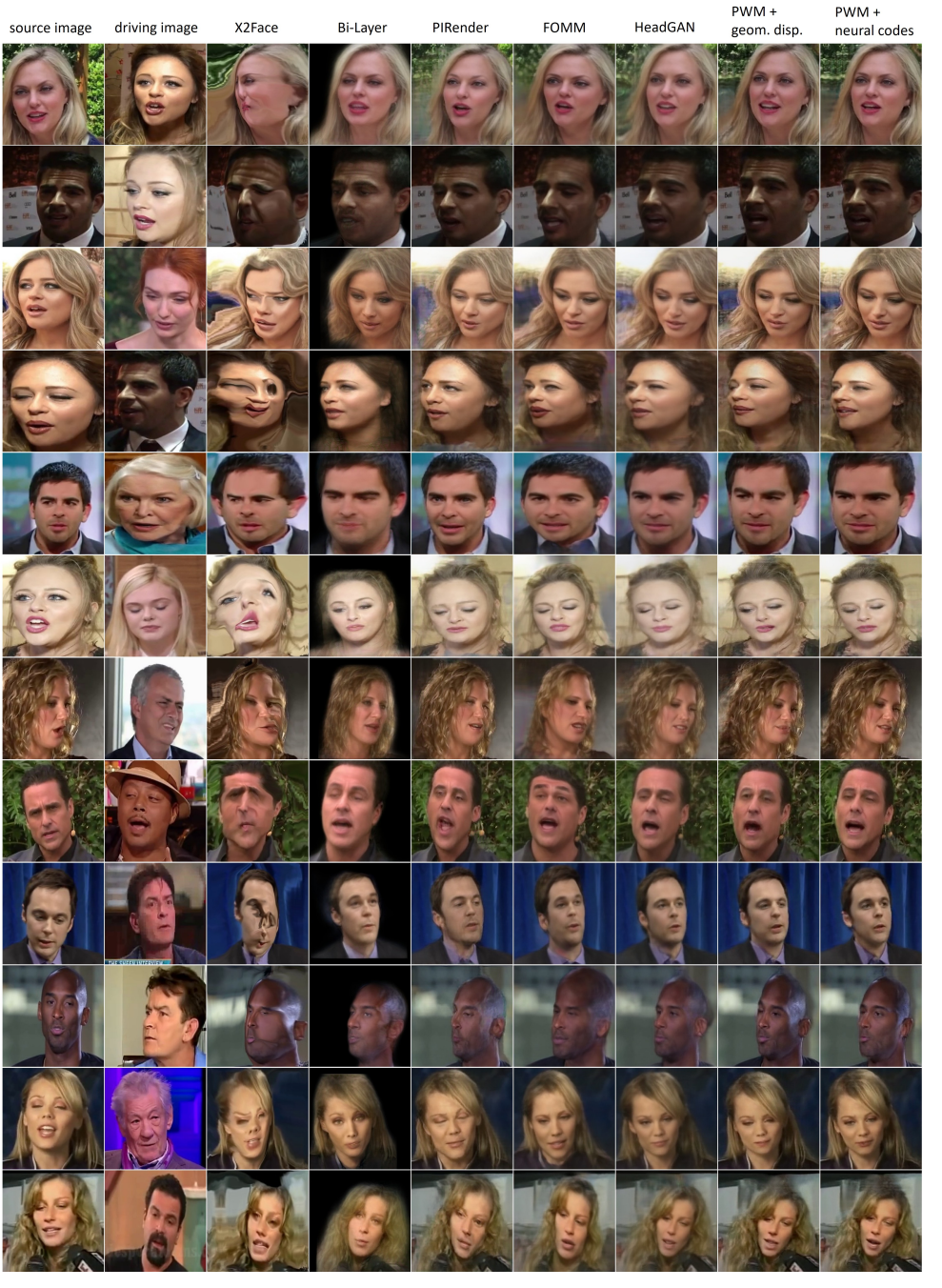

Figure 3: Qualitative results for cross identity reconstruction.

References

[1] face alignment. URL https://github.com/1adrianb/face-alignment.

[2] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017.

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[7] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.