

Supplementary Material: Sparse to Dense Motion Transfer for Face Image Animation

Ruiqi Zhao^{1,2}, Tianyi Wu^{1,2}, Guodong Guo^{1,2}

¹Institute of Deep Learning, Baidu Research, Beijing, China

²National Engineering Laboratory for Deep Learning Technology and Application, Beijing, China

{zhaoruiqi, wutianyi01, guogudong01}@baidu.com

In this supplementary material, we provide model implementation details in Section 1 and training details in Section 2.

1. Network Architecture Details

Dense motion estimation network. For the global motion estimation network, the input image is the 256×256 source image. It has an encoder-decoder architecture with seven $conv_{3 \times 3} - bn - relu - avg_pool_{2 \times 2}$ down-sampling blocks in the encoder and seven $upsample_{2 \times 2} - conv_{3 \times 3} - bn - relu$ up-sampling blocks in the decoder. The last four blocks of the encoder and the first four blocks of the decoder are connected to source landmark vector S_{lm} and driving landmark vector D_{lm}^i respectively through AdaIN layer [3]. The hidden representation is connected to the vector $D_{lm}^i - S_{lm}$ through Add_Motion layer, Figure 1. The seven blocks of the encoder have 32, 64, 128, 256, 512, 1024 and 1024 feature maps respectively. And the seven blocks of the decoder have 1024, 1024, 512, 256, 128, 64 and 32 feature maps respectively. For each of the three small local motion estimation networks, the input image is a 64×64 image patch centered at the left eye eyebrow, the right eye eyebrow and the mouth area respectively. As the global motion estimation network, the local motion estimation network also has the encoder-decoder architecture with five down-sampling blocks in the encoder and five up-sampling blocks in the decoder. The last three blocks of the encoder and the first three blocks of the decoder are controlled by local source landmarks vector and local driving landmarks vector through AdaIN layer. The five blocks of the encoder have 16, 32, 64, 128 and 256 feature maps respectively. And the five blocks of the decoder have 256, 128, 64, 32 and 16 feature maps respectively. Note the input of encoders in the motion estimation network only depend on the source image/landmarks. Therefore during inference they only need to run once and the computation cost only depends on the decoders.

Image generation network. We use the same architec-

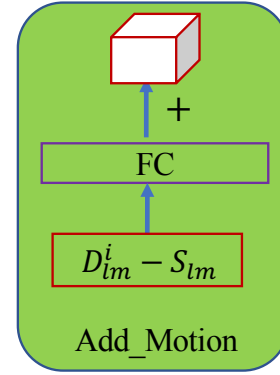


Figure 1. Add_Motion layer

ture as in [5]. It adopts the Johnson architecture [2] with one image filtering block, two down-sampling blocks, six residual blocks, two up-sampling blocks and one image filtering block.

Discriminator network. We use the PatchGAN [1] network architecture. It has four $conv_{4 \times 4} - bn - relu - avg_pool_{2 \times 2}$ down-sampling blocks and one final 1×1 convolutional layer. The four down-sampling blocks have 128, 256, 512 and 1024 feature maps respectively.

2. Training Details

For VoxCeleb1 dataset, we first train 140 epochs without the GAN loss. In each epoch, for each video we randomly select two images, one image is used as source image and the other image is used as driving image. We repeat 75 times for each video. We use the Adam [4] optimizer with initial learning rate 2×10^{-4} for both the dense motion estimation network and the image generation network. We drop learning rate after training for 60 and 80 epochs. We use batch size 40 on 4 Tesla V100-16GB gpus for training. After it converges in approximately 25 hours, we add adversarial loss and do further training. We use initial learn-

ing rate 2×10^{-4} for both the dense motion estimation network and the image generation network and 2×10^{-6} for the discriminator. We drop learning rate after training for 60 and 80 epochs. The training takes another 42 hours to converge. The training setting we use for FaceForensics dataset is similar to VoxCeleb1 dataset except the number of training epochs is 80.

References

- [1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.
- [2] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.
- [3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.
- [4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.
- [5] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019.