Supplemental Materials: A Generative Model of Worldwide Facial Appearance

Zachary Bessinger Nathan Jacobs
Department of Computer Science
University of Kentucky
{zach, jacobs}@cs.uky.edu

1. Network Details

A complete description of the *GPS2Face* network architecture is shown in Table 1. All networks are implemented in PyTorch 0.3.1. All internal nodes of each network use LeakyReLU activations, $\alpha = 0.1$, except for the final layer of D_z and D_{img} which use sigmoid activations and G which uses tanh. Batch normalization (BN) is applied prior to the activation function where "BN" is present. The layer name "PS Conv" refers to a *PixelShuffle* convolution [2].

We first pre-train the landmark network by minimizing the Huber loss [1]:

$$\begin{aligned}\mathcal{L}_{\text{huber}}(x, y) &= \frac{1}{n}\sum_i z_i \\ z_i &= \begin{cases} 0.5(x_i - y_i)^2, & \text{if } |x_i - y_i| < 1 \\ |x_i - y_i| - 0.5, & \text{otherwise} \end{cases},\end{aligned} \tag{1}$$

using Adam with a learning rate of 0.01, $\beta_1 = 0.5$, and $\beta_2 = 0.999$ until convergence. The output layer has 136 neurons, the flattened set of 68 x, y pairs of landmarks.

We then train the image generation component of *GPS2Face* using Adam with batch sizes of 64. It is optimized for 100 000 iterations with a learning rate of 0.0001, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. The dimension of the latent space, z, (output of the Encoder network) is set to 50. The latent factors, $\boldsymbol{c} = \{$age, gender, country code, latitude/longitude, and pose angles$\}$, and facial landmarks, s, are concatenated on the channel axis of each respective network feature. Age, gender, and country code are one-hot encoded. Conditioning terms on z are concatenated on the first axis. Conditioning terms in D_x are replicated then concatenated along the channel axis.

Table 1: Detailed network architecture.

Layer	**Kernel/Stride/Pad**	**Out Shape**
Conv1	5×5, 2, 2	64×64 ×128
Conv2	5×5, 2, 2	32×32×256
Conv3	5×5, 2, 2	16×16×512
Conv4	5×5, 2, 2	8×8×1024
Linear1		50

(a) Encoder, E

Layer	**Kernel/Stride/Pad**	**Out Shape**
Linear1/BN		64
Linear2/BN		32
Linear3/BN		16
Linear4		1

(b) Discriminator z, D_z

Layer	**Kernel/Stride/Pad**	**Out Shape**
Concat(z, c, s)		
Linear1		8×8×1024
PS Conv1	3×3, 1, 1	16×16×512
PS Conv2	3×3, 1, 1	32×32×256
PS Conv3	3×3, 1, 1	64×64×128
PS Conv4	3×3, 1, 1	128×128×64
Conv5	3×3, 1, 1	128×128×3

(c) Generator, G

Layer	**Kernel/Stride/Pad**	**Out Shape**
Conv1	5×5, 2, 2	64×64×16
Concat(Conv1, c, s)		
Conv2/BN	5×5, 2, 2	32×32×32
Conv3/BN	5×5, 2, 2	16×16×64
Conv4/BN	5×5, 2, 2	8×8×128
Linear1		1024
Linear2		1

(d) Discriminator Image, D_x

Layer	**Kernel/Stride/Pad**	**Out Shape**
Linear1		256
Linear2		256
Linear3		136

(e) Landmark Network, L

References

[1] R. Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision*, 2015.

[2] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.