Photo-realistic Monocular Gaze Redirection Using Generative Adversarial Networks – Supplementary Material –

Zhe He[1, 2], Adrian Spurr[1], Xucong Zhang[1], Otmar Hilliges[1]
[1]AIT Lab, ETH Zürich
[2]Institute of Neuroinformatics, ETH Zürich & University of Zürich
zhehe@student.ethz.ch, {adrian.spurr, xucong.zhang, otmar.hilliges}@inf.ethz.ch

1. Network Architecture

In this section, we provide the details of network architecture discussed in Sec. 4.1.

1.1. Abbreviations

Conv($k \times k$, s, p): A convolutional layer with kernel size $k \times k$, stride size s and padding size p. Zero padding is used in all convolutional layers. **IN**: An instance normalization layer. **ReLU**: A ReLU activation layer. **LReLU**: A Leaky ReLU activation layer. Slope of the activation function at $x < 0$ is set to 0.01. **Tanh**: A tanh activation layer. **DeConv**($k \times k$, s, p): A transposed convolutional layer with kernel size $k \times k$, stride size s and padding size p. Zero padding is used in all transposed convolutional layers. **Res**($k \times k$, s, p, IN, ReLU): A residual layer which builds upon Conv($k \times k$, s, p), IN and ReLU layers.

1.2. Generator

Layers	Output
Conv(7x7, 1, 3)–IN–ReLU	(64, 64, 64)
Conv(4x4, 2, 1)–IN–ReLU	(32, 32, 128)
Conv(4x4, 2, 1)–IN–ReLU	(16, 16, 256)
Res(3x3, 1, 1, IN, ReLU)	(16, 16, 256)
Res(3x3, 1, 1, IN, ReLU)	(16, 16, 256)
Res(3x3, 1, 1, IN, ReLU)	(16, 16, 256)
Res(3x3, 1, 1, IN, ReLU)	(16, 16, 256)
Res(3x3, 1, 1, IN, ReLU)	(16, 16, 256)
Res(3x3, 1, 1, IN, ReLU)	(16, 16, 256)
DeConv(4x4, 2, 1)–IN–ReLU	(32, 32, 128)
DeConv(4x4, 2, 1)–IN–ReLU	(64, 64, 64)
Conv(7x7, 1, 3)–Tanh	(64, 64, 3)

Table 1: Generator Architecture

1.3. Discriminator

Layers	Output
Conv(4x4, 2, 1)–LReLU	(32, 32, 64)
Conv(4x4, 2, 1)–LReLU	(16, 16, 128)
Conv(4x4, 2, 1)–LReLU	(8, 8, 256)
Conv(4x4, 2, 1)–LReLU	(4, 4, 512)
Conv(4x4, 2, 1)–LReLU	(2, 2, 1024)

Table 2: Backbone Network of Discriminator

Layers	Output
Backbone	(2, 2, 1024)
Conv(2x2, 1, 1)	(3, 3, 1)

Table 3: Discriminator Architecture

Layers	Output
Backbone	(2, 2, 1024)
Conv(2x2, 1, 0)	(1, 1, 2)

Table 4: Gaze Estimator Architecture

2. Implementation

Code for training and testing our model is available online (https://github.com/HzDmS/gaze_redirection).

3. Training Details of Gaze Estimators

Training details of the gaze estimators used in Sec. 5.6 are provided in this section. We used the Adam optimizer with learning rate 0.00005, $\beta_1 = 0.5$, and $\beta_2 = 0.999$. Batch size was set to 32. For the training on the raw dataset, the gaze estimator was trained for 200 epochs. For the training on the augmented dataset, the gaze estimator was trained for 100 epochs.

4. Results on Non-frontal Faces

We conducted an additional experiment on non-frontal head poses and compared them with the frontal head pose. We used the same settings as introduced in Sec. 4.2 and Sec. 5.2 (in our paper). Samples which could not be successfully parsed with `dlib` [1] were not included in the training and test datasets. Note this process removed some samples with extreme head poses.

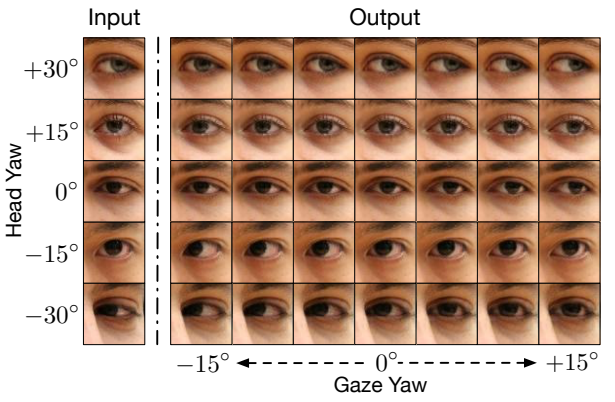

Figure 1: Gaze redirection results on images with different head poses. In the output images, the gaze pitch is equal to 0°

Fig. 1 shows redirected eye-images (with 0° output gaze pitch) using input images with varying head-poses. The method produces high-quality results on these inputs.

Using the evaluation protocol and metrics introduced in Sec. 5.1 and Sec. 5.3 (in our paper), Fig. 2(a) shows that the LPIPS scores of the generated images are consistent up to ±15°. The LPIPS scores for larger head angles (±30°) are worse than the ones of (0°, ±15°). We note that: 1) There are fewer training samples with large head poses due to `dlib` detection failures. 2) These samples are more difficult in general, due to self-occlusion under extreme viewing angles. For example, in the input of the bottom row (Fig. 1), the eye-corner is completely occluded by the nose.

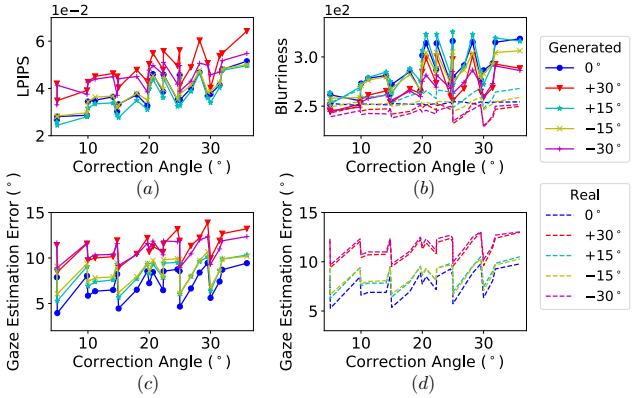

Figure 2: Quantitative evaluation results

The blurriness scores in Fig. 2(b) indicate that head pose only marginally affect image sharpness.

Fig. 2(c) shows that large head poses lead to large gaze estimation error for our generated images. Comparing Fig. 2(c) and (d) shows that the gaze estimation error of generated and real images with the same head angle are consistent with each other. It suggests that the generated images are of similar quality to real ones wrt to the gaze estimation task. In summary, this experiment provides evidence that the proposed method performs well, even on eye images generated with different head poses.

References

[1] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009. 2