AutoRetouch: Automatic Professional Face Retouching Supplementary Material

Alireza Shafaei
Skylab Technologies Inc.†
The University of British Columbia
alireza@skylabtech.ai

James J. Little
The University of British Columbia
little@cs.ubc.ca

Mark Schmidt
The University of British Columbia
CCAI Affiliate Chair (Amii)
schmidtm@cs.ubc.ca

Contents

1. Retouching

1.1. The Discriminator

For the discriminator we use the following sequential architecture from PatchGAN [1].

```
(0): Conv2d(3, 64, kernel_size=(4, 4),
            stride=(2, 2), padding=(2, 2))
(1): LeakyReLU(negative_slope=0.2, inplace)
(2): Conv2d(64, 128, kernel_size=(4, 4),
            stride=(2, 2), padding=(2, 2))
(3): InstanceNorm2d(128, eps=1e-05, momentum=0.1,
            affine=False)
(4): LeakyReLU(negative_slope=0.2, inplace)
(5): Conv2d(128, 256, kernel_size=(4, 4),
            stride=(2, 2), padding=(2, 2))
(6): InstanceNorm2d(256, eps=1e-05, momentum=0.1,
            affine=False)
(7): LeakyReLU(negative_slope=0.2, inplace)
(8): Conv2d(256, 512, kernel_size=(4, 4),
            stride=(1, 1), padding=(2, 2))
(9): InstanceNorm2d(512, eps=1e-05, momentum=0.1,
            affine=False)
(10): LeakyReLU(negative_slope=0.2, inplace)
(11): Conv2d(512, 1, kernel_size=(4, 4),
            stride=(1, 1), padding=(2, 2))
```

1.2. Multiscale Patch Sampling

The number of $w \times w$ patches in the image increases quadratically with respect to the scale of the image. We wish to augment the training data by downsampling the images. If we choose a down-scaling parameter at uniform, it will under-represent the $w \times w$ patches in the spatially larger images. Therefore we should be sampling higher-resolution images quadratically more often to make all the patches equally likely. The cumulative distribution function of a quadratically increasing density function would be cubic. Using the inverse transform sampling method we can simply use $s^{\frac{1}{3}}$ as the scaling factor, where $s \sim \mathrm{Unif}(0, 1)$. This sampling procedure is approximate because we also have a minimum scale that truncates the density function. If we set a minimum scale $s_{\min}$, the scaling factor would become $(s(1 - s_{\min}^3))^{\frac{1}{3}}$; however, since we are using high-resolution images, the minimum scaling factor cubed $s_{\min}^3$ will become small enough ($< 10^{-3}$) that the difference would become negligible. However, if the training data is not high-resolution, omitting the extra term could still produce unbalanced samples.

1.3. User Study

To perform the user study, we developed the UI in Fig. 1. Each time, we show the user the output of two algorithms in random order and ask them to pick their favourite. At the bottom of the page, we show the user a zoomed-in version of the original image and the two outputs and an image highlighting the differences. As the users move the mouse cursor over the original image, they can visually inspect and compare a zoomed version of all the images. To conduct our user studies, we hired three professional retouchers. We run several experiments under this evaluation framework and present the results in the main paper.

For FFHQR evaluations, we pick 1000 images from the test set. More specifically, we use the images from `63000` to `63999`. Similarly, for the studio data, we pick 1000 images from the test set. Each user sees the 1000 evaluations in random order. We ran seven experiments, collected over 5400 votes, while spending 48 hrs in total.

† The author produced the results at Skylab Technologies Inc.

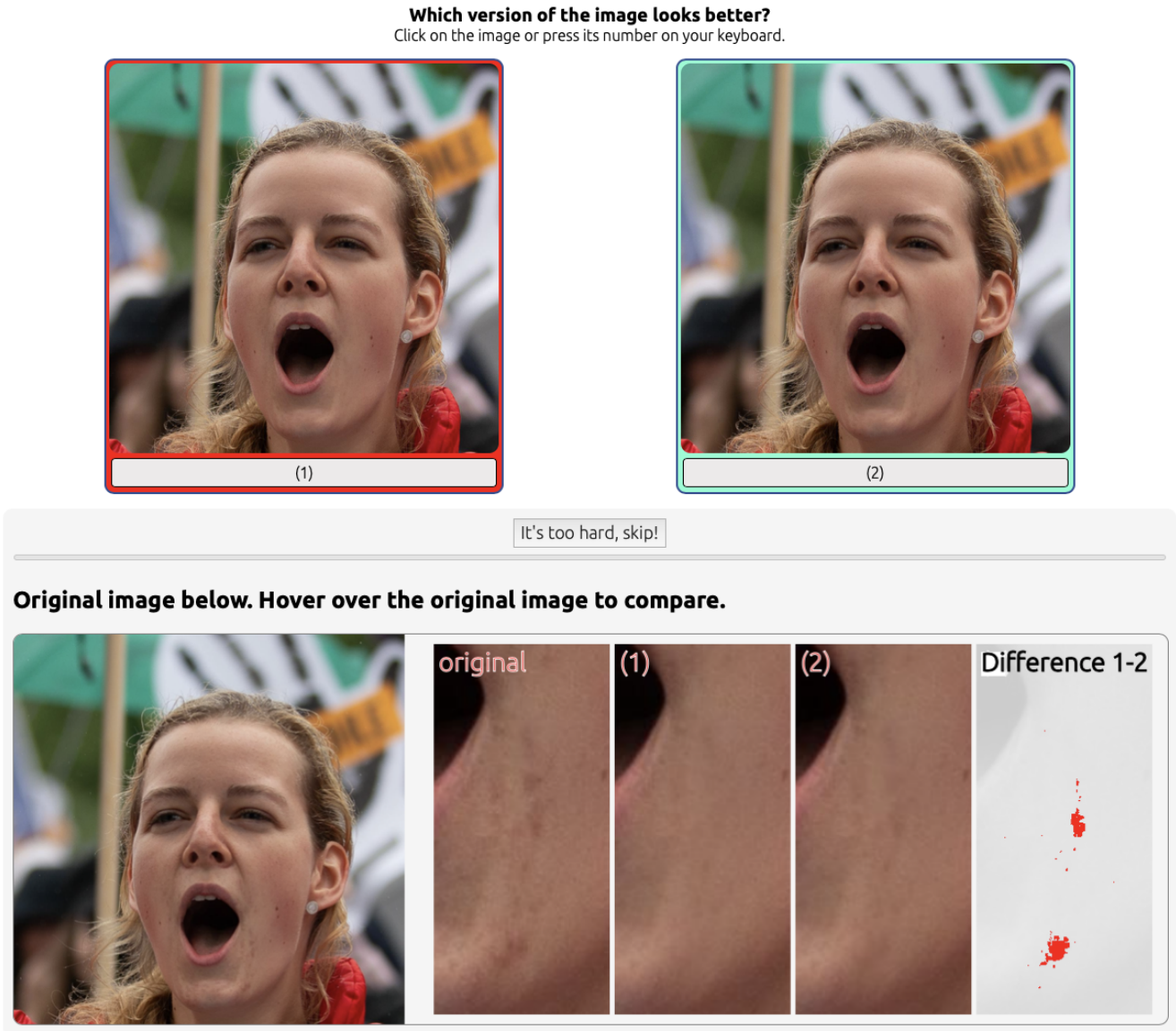

Figure 1: The user study UI. The users are shown two images in random order, and they will decide which version they prefer. At the bottom of the page, dynamically changing figures allow easy comparison between the algorithms.

1.4. Evaluation

Figure 2a compares the retouching output of our model with the groundtruth patches. Our model preserves the fine texture better than the groundtruth data. In the main paper we perform ablation studies on the effect of each term in the loss function. We observed that adding RAGAN loss term encourages the model to preserve the input as much as possible. The preservation of the details produces retouching models that perform better than the groundtruth retouching data. We suspect this happens because, in real-life retouching, the professionals may use large brushes for correction that would inadvertently affect areas of the image that do not require retouching. Our model, however, operates at the pixel level and can preserve details at no extra cost.

We also test our retouching model on lower-resolution smartphone camera images. The images are captured with iPhone 6 or iPhone X. Figure 2b shows sample outputs from our tests.

Figure 3a shows the failure cases of our retouching model. The images that our model fails to correct usually contain severe skin blemishes. Although our model improves the output image, it does not fully eliminate the blemishes.

2. Dataset

2.1. FFHQR

See Fig. 4 for more samples of our new retouching dataset.

2.2. Studio Data

Figure 5 shows the distribution of the head-crop images that we extracted from the studio training data to train our models.

References

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

(a) (b)

Figure 2: (a) The output of our model compared to the groundtruth retouching. The left column is the input, the middle column is groundtruth retouching, and the right column is our output. Our model preserves the fine details more than the groundtruth. (b) Sample input/output of retouched images captured with cellphones. The figure is best viewed on a screen.

(a) Failure cases. Our retouching model fails when the blemishes are severe.

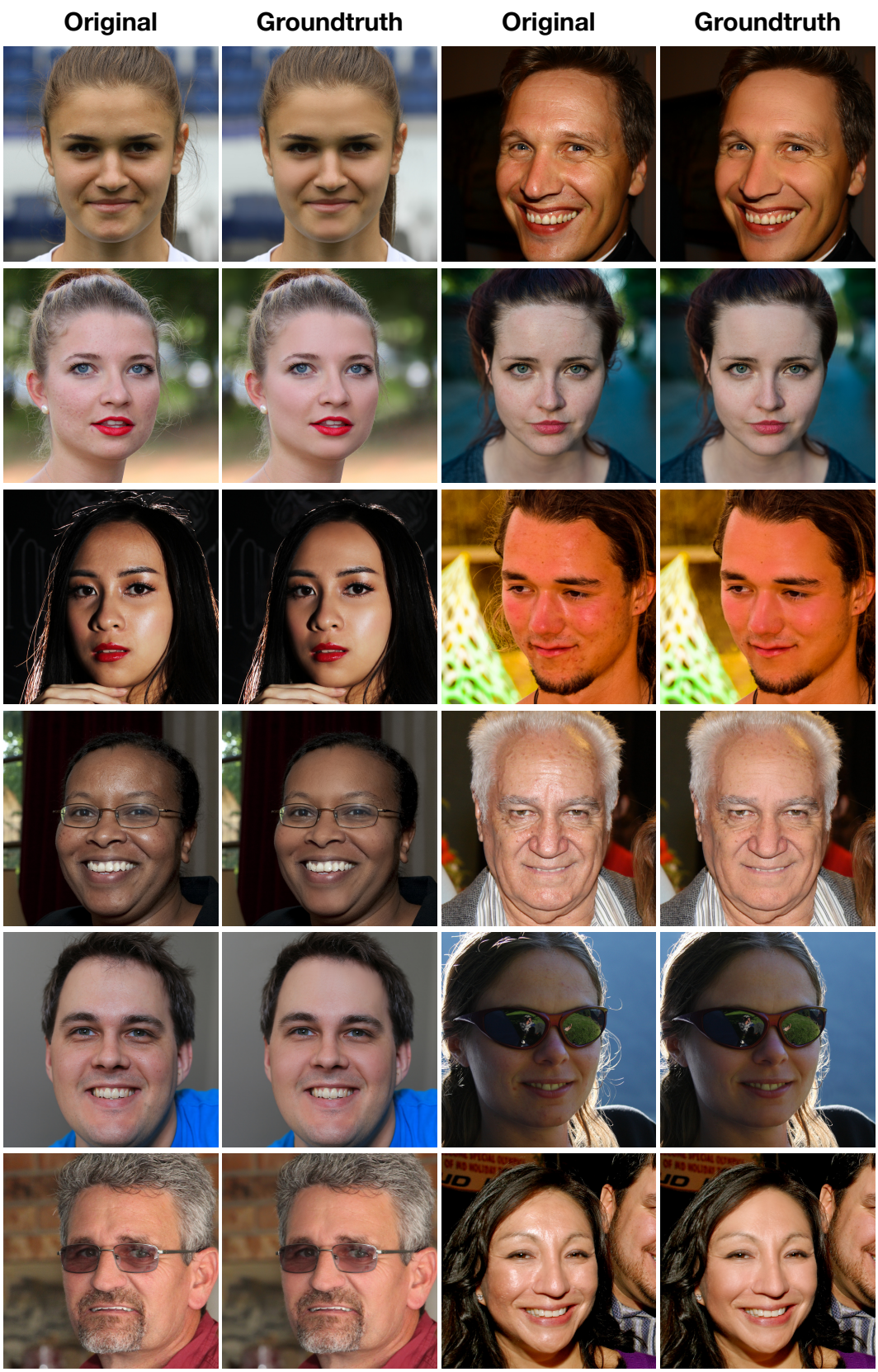

Figure 4: More samples from our new retouching dataset FFHQR.

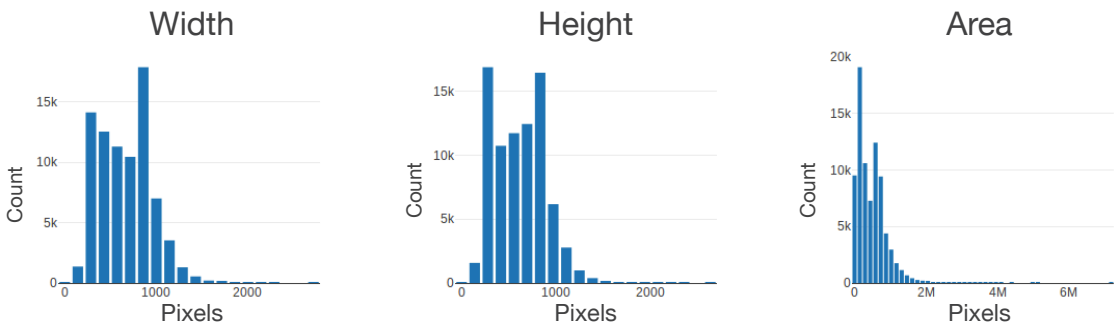

Figure 5: The distribution of width, height, and area of the head-crops extracted from the studio retouching data. The data is similar to FFHQR in resolution.