Debiasing Image-to-Image Translation Models - Supplementary Material

Md Mehrab Tanjim[1]
mtanjim@eng.ucsd.edu
Krishna Kumar Singh[2]
krishsin@adobe.com
Kushal Kafle[2]
kkafle@adobe.com
Ritwik Sinha[2]
risinha@adobe.com
Garrison W. Cottrell[1]
gary@eng.ucsd.edu

[1] University of California San Diego
9500 Gilman Dr,
La Jolla, CA, USA

[2] Adobe Research
345 Park Ave,
San Jose, CA, USA

1 Train-Validation-Test Split CelebA-HQ

For experimentation with the CelebA-HQ [3] dataset, we make separate datasets for each of the attributes, namely 'Eyeglasses', 'Bald', and 'Wearing Hat.' For fair comparison, we hold out an equal number of images from majority and minority classes for both validation and testing. Note that the number of images from the minority classes are quite low in number (e.g., 700 images for 'Bald'). Therefore, we take 50 images from each class for validation (100 in total) and 150 images for the test set (total 300 images). From the rest of the images, we adjust the training dataset such that the original bias ratio remains unchanged for our selected attributes. The exact number of images for each of the training dataset along with validation and test is given in Table 1.

Attribute	Group	Train	Validation	Test
Bald	Minority	512	50	150
	Majority	21,062	50	150
	Both	21,574	100	300
Wearing Hat	Minority	870	50	150
	Majority	23,523	50	150
	Both	24,393	100	300
Eyeglasses	Minority	1,268	50	150
	Majority	24,645	50	150
	Both	25,913	100	300

Table 1: Train-validation-test splits for specific attributes.

2 Training Procedure

Data Augmentations. We apply data augmentations to both majority and minority classes. Specifically, these augmentations are shifting 10% (both vertically and horizontally), shear-

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

Figure 1: Example cases of bias for both super-resolution and sketch-to-face. Here, for both tasks, the attributes are visible in the inputs images (i.e. Eyeglasses, Hat, Baldness) but they are missing in the generated images.

ing 10%, scaling 10%, and mirror flipping.

Classifiers. For training, we replace the last fully connected layer of a pretrained ResNet50 [2], and re-train it again on our training dataset. This classifier is used for applying auxiliary classifier loss, $\mathcal{L}_c$. We train three separate classifiers, one for each attribute. Their performances on the test sets are as follows:

- *Bald.* F1 score: 0.8889, Prediction scores: 87.13%, Accuracy: 90%
- *Eyeglasses.* F1 score: 0.9899, Prediction scores: 95.99%, Accuracy: 99%
- *Wearing Hat.* F1 score: 0.951, Prediction scores: 90.65%, Accuracy: 95.33%

To train the classifier for evaluation, we train a deeper network, ResNet152 [2], on the same training sets. This improves the accuracy for prediction for 'Bald' (94%) and 'Eyeglasses'(99.33%), which makes it reliable for prediction. This also keeps our evaluation classifier network architecture and weights separate from ResNet50 network which was used for $\mathcal{L}_c$ during training. This is critical for fair evaluation as we do not want to evaluate using a classifier which was used for training since the network is optimized to perform well for it. Training the evaluation classifier follows the same procedure. Their performances are:

- *Bald.* F1 score: 0.9362, Prediction scores: 90.16%, Accuracy: 94%
- *Eyeglasses.* F1 score: 0.9933, Prediction scores: 94.67%, Accuracy: 99.33%
- *Wearing Hat.* F1 score: 0.951, Prediction scores: 91.44%, Accuracy: 95.33%

Measuring Bias. We use CelebA-HQ [3] to demonstrate how we can measure bias in image-to-image translation tasks. We use CelebA-HQ for two reasons. First, it has 40 labeled binary attributes (e.g. 'Eyeglasses', 'Bald', etc.), making detecting bias easier compared to unlabeled data (such as Flicker-Face-HQ or FFHQ [4]). Second, it is widely used to train both conditional and unconditional generative models [1, 3, 5, 7, 8], making it an ideal dataset for bias analysis. Some additional examples of biases are showed in Figure 1.

For measuring bias, we generate images for the super-resolution task. We down-sampled the ground truth test images by a factor of 8 and applied the super-resolution pSp network (which has been pre-trained on the same dataset as our classifier). We use a ResNet152 classifier (trained on pSp training set) to calculate the F1 scores on real and generated images for measuring biases. These numbers are reported in Table 1 in the *main* paper. Note that

Attribute	Bald	Wearing Hat	Eyeglasses	Blond Hair	Bangs	Black Hair	Male	Heavy Makeup	High Cheekbones	Smiling
Percentage	**2.37**	**3.57**	**4.89**	17.09	18.08	21.97	36.86	45.69	46.16	46.97
Low-Resolution Inputs	0.6488	0.8523	0.6613	0.5891	0.6280	0.8173	0.9414	0.2988	0.4641	0.7595
Sketches	0.2234	0.7215	0.8747	0.0000	0.3366	0.0311	0.7615	0.5590	0.1606	0.5848

Table 2: F1 scores on low-resolution and sketch input images.

having a low F1 score means either 1) low recall: the model failed to generate the desired attribute in the generated images (for example, not generating images with eyeglasses where the ground truth images have this attribute), or 2) low precision: the model generated an attribute where it should not have (e.g., producing hair when the person is bald). Both cases represent a bias problem. Therefore, low F1 score on the generated images can reveal the biases for the attribute.

Here, we report the F1 scores on the low resolution inputs and sketches. The numbers appear in Table 2. The numbers for most of the attributes, especially for 'Eyeglasses' and 'Wearing Hat', show that there is enough information in the input images to generate the attributes in question.

Training. Here we describe our changes to Pixel2Style2Pixel [7] (pSp) and pix2pix. In the case of pSp, it encodes the input images using a feature pyramid backbone [6] and maps them to the extended latent space of a frozen StyleGAN2 [5] generator (pre-trained on FFHQ [4]), $\mathcal{W}+$, which is consists of 18 different 512-dimensional feature vectors, one for each StyleGAN2 layer. We apply our contrastive loss (Equation 1 in the main paper), $\mathcal{L}_s$, to the latent codes of each of the layers in $\mathcal{W}+$ separately. The latent codes then are followed by MLP layers, which consists of two linear feed-forward networks with 512 hidden units and a ReLU activation in between (one MLP for each of 18 input layers in StyleGAN2). The temparature parameter in Equation 1 is set to 0.07 for all experiments. Finally, we apply auxiliary classifier loss (Equation 2), $\mathcal{L}_c$, on the outputs of the decoder.

For pix2pix, we apply the U-Net architecture for the sketch-to-image translation model. Similar to pSp, we refer to the original pix2pix model as 'Vanilla' and re-sampling the minority during training as 'Sampling Baseline.' For our model (I+II+III), we make similar changes to pix2pix. Specifically, on top of re-sampling, we apply the supervised contrastive loss, $\mathcal{L}_s$ (Equation 1) to the output of bottleneck layer of the encoder. After applying $\mathcal{L}_s$, we pass the features through MLP layer, ϕ, and add an auxiliary classifier loss (Equation 2), $\mathcal{L}_c$, at the end. We also experiment on other two variations of our model (I+II and I+III).

For training our model for both pSp and pix2pix, we follow a curriculum learning procedure. We introduce our losses (supervised contrastive loss, $\mathcal{L}_s$, and auxiliary classifier loss, $\mathcal{L}_c$) after k iterations. We start with a small value of m for both hyperparameters for supervised contrastive loss, λ_s, and auxiliary classifier loss, λ_c. These hyperparameters are then increased by m every k iterations.

For pSp, $k = 10,000$, and $m = 0.001$. We apply similar curriculum training procedure for pix2pix as well. Here, $m = 0.01$. Instead of k iterations, we apply the losses after the first epoch. The values of the hyperparameters are increased by the same value, m, after each epoch. For both pSp and pix2pix, Ours (I+II) and (I+III) follow the same training steps, except the hyperparameter for a specific loss is set to zero. For example, for Ours (I+II), $\lambda_c = 0$, and for Ours (I+III), $\lambda_s = 0$.

3 Additional Results and Examples

We show additional examples for the Vanilla and Our Model (I+II+III) from our human face experiments in Figure 2-7.

Figure 2: Results for super-resolution task on ‘Eyeglasses’.

Figure 3: Results for super-resolution task on ‘Bald’.

Figure 4: Results for super-resolution task on 'Wearing Hat'.

Figure 5: Results for sketch-to-face task on 'Eyeglasses'.

Figure 6: Results for sketch-to-face task on ‘Bald’.

Figure 7: Results for sketch-to-face task on ‘Wearing Hat’.

References

[1] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8188–8197, 2020.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[6] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[7] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2287–2296, 2021.

[8] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.