Facial Attribute Transformers for Precise and Robust Makeup Transfer Supplementary Materials

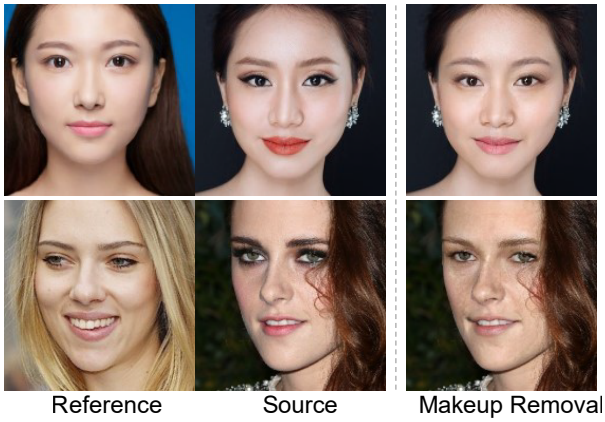

Figure 1. Makeup removal examples.

A. Cycle-GAN Training in Makeup Transfer

As stated in the paper, Cycle-GANs enable training with unpaired data and are the common practice in makeup transfer. Following we provide detailed training objectives of Cycle-GANs, in terms of makeup transfer.

We start from the aggregation of loss functions provided in the paper:

$$\begin{aligned} J_D^{adv} &= -\mathbb{E}_{x\sim\mathcal{P}_X}\left[\log D_X(x)\right] - \mathbb{E}_{y\sim\mathcal{P}_Y}\left[\log D_Y(y)\right] \\ &\quad - \mathbb{E}_{x\sim\mathcal{P}_X,y\sim\mathcal{P}_Y}\left[\log\left(1 - D_X(G(y,x))\right)\right] \\ &\quad - \mathbb{E}_{x\sim\mathcal{P}_X,y\sim\mathcal{P}_Y}\left[\log\left(1 - D_Y(G(x,y))\right)\right] \\ J_G &= \lambda_{adv}J_G^{adv} + \lambda_{cyc}J_G^{cyc} + \lambda_{per}J_G^{per} + \lambda_{make}J_G^{make}, \end{aligned} \tag{1}$$

where J_D^{adv} and J_G are the loss for discriminators and the generator, respectively. Note that J_D^{adv} is formed with two groups of terms, which corresponds two discriminators in makeup transfer: One is for discriminating the generated makeup face from the reference, and the other is oriented at distinguishing the generated non-makeup face from the source. The formulation indicates a clear message, that the removal of makeup is intrinsically trained along with the task of makeup transfer as shown in Fig. 1.

Then let us look into the loss of the generator in Eq. 1, where we add J_G^{make} indicating makeup loss. The adversarial loss of generator is given in the regular form of adversarial training:

$$\begin{aligned} J_G^{adv} &= -\mathbb{E}_{x\sim\mathcal{P}_X,y\sim\mathcal{P}_Y}\left[\log D_X(G(y,x))\right] \\ &\quad - \mathbb{E}_{x\sim\mathcal{P}_X,y\sim\mathcal{P}_Y}\left[\log D_Y(G(x,y))\right] \end{aligned} \tag{2}$$

The key that Cycle-GAN can be trained without paired data is laid in the cycle consistency loss [3]. L1 loss is used to supervise the reconstructed source from the generated face:

$$\begin{aligned} J_G^{cyc} &= \mathbb{E}_{x\sim\mathcal{P}_X,y\sim\mathcal{P}_Y}\left[\|G(G(x,y),x) - x\|_1\right] \\ &\quad + \mathbb{E}_{x\sim\mathcal{P}_X,y\sim\mathcal{P}_Y}\left[\|G(G(y,x),y) - y\|_1\right]. \end{aligned} \tag{3}$$

For preserving the identity and perceptual details of source, A VGG-16 CNN is adopted to keep the consistency of extracted features. The perceptual loss is formulated as:

$$\begin{aligned} J_G^{per} &= \mathbb{E}_{x\sim\mathcal{P}_X,y\sim\mathcal{P}_Y}\left[\|F_l(G(x,y)) - F_l(x)\|_2\right] \\ &\quad + \mathbb{E}_{x\sim\mathcal{P}_X,y\sim\mathcal{P}_Y}\left[\|F_l(G(y,x)) - F_l(y)\|_2\right], \end{aligned} \tag{4}$$

where F_l indicates the lth layer feature of the pre-trained VGG.

B. Grid Generation of TPS

Similar to STN [1], the Spatial FAT uses a grid generator to compute a sampling grid $\mathcal{P} = \{p_i\}$ on an image to form a transformation. The 2D TPS transformation we introduce into Spatial FAT is parameterized by a $2 \times (K+3)$ matrix [2]:

$$T = \begin{bmatrix} a_0, a_1, a_2, u \\ b_0, b_1, b_2, v \end{bmatrix}, \tag{5}$$

where $u, v \in R^{1\times K}$. **Following we use the formulation from [2] to describe the grid computation of 2D TPS.** For a point $p \in R^{1\times 2}$, its sampling point is computed by a linear projection:

$$p' = T \begin{bmatrix} 1 \\ p \\ \phi(\|p - c_1\|) \\ \vdots \\ \phi(\|p - c_K\|), \end{bmatrix} \tag{6}$$

where $\phi(r) = r^2 log(r)$ as the radial basis kernel applied to the Euclidean distance between p and control points C. Then the problem is to solve a linear system to find the coefficients of TPS:

$$c_i' = T \begin{bmatrix} 1 \\ p \\ \phi(\|c_i - c_1\|) \\ \vdots \\ \phi(\|c_i - c_K\|), \end{bmatrix}, i, \ldots, K, \tag{7}$$

subject to boundary conditions:

$$\begin{aligned} 0 &= u\mathbf{1} \\ 0 &= v\mathbf{1} \\ 0 &= uC_x^T \\ 0 &= vC_y^T, \end{aligned} \tag{8}$$

where C_x and C_y are the first and second dimension coordinates, respectively. In a matrix form, T has a closed-form solution.

$$T = \left[C', \mathbf{0}^{2\times 3}\right] \Delta_C^{-1}. \tag{9}$$

Both the solving and application of TPS can be integrated into the neural networks, since they are differentiable matrix operations.

C. Implementation of GT Generation

To help reproducing, we provide the GT generation implementation here, which requires advanced skills to maintain efficiency, and will release other code with the acceptance of the paper.

References

[1] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2017–2025, 2015.

[2] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048, 2018.

[3] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.