

Supplementary Material

Chaeyeon Chung^{*1,2}
cy_chung@kaist.ac.kr

Taewoo Kim^{*1,2}
specia1ktu@kaist.ac.kr

Hyelin Nam^{*3}
nhl0208@cau.ac.kr

Seunghwan Choi¹
shadow2496@kaist.ac.kr

Gyojung Gu¹
gyojung.gu@kaist.ac.kr

Sunghyun Park^{1,2}
psh01087@kaist.ac.kr

Jaegul Choo¹
jchoo@kaist.ac.kr

¹ KAIST AI
South Korea

² Kakao Enterprise
South Korea

³ Chung-Ang University
South Korea

1 Model Architecture and Implementation Details

1.1 Flow-based Hair Alignment

As described in Section 3.2 of our paper, the flow-based hair alignment module aligns the target hairstyle with the source pose using a dense optical flow estimated by the hair flow estimation network. We obtain the dense optical flow map $\mathcal{F} \in \mathbb{R}^{2 \times H \times W}$ by combining a coarse keypoint difference and the refinement flow map $\mathcal{F}_{ref} \in \mathbb{R}^{2 \times H \times W}$.

To be specific, our module first converts facial keypoints $\mathbf{k} \in \mathbb{R}^{N_k \times 2}$ into the Gaussian keypoint heatmap $\mathbf{H} \in \mathbb{R}^{N_k \times H \times W}$, where N_k denotes the number of keypoints. Then, we obtain the keypoint heatmap difference $\hat{\mathbf{H}} \in \mathbb{R}^{N_k \times H \times W}$, which is calculated as $\mathbf{H}_t - \mathbf{H}_s$, where t and s indicate a target and a source, respectively. With $\hat{\mathbf{H}}$ and the warped \mathbf{I}_t , the flow estimation network, FE , predicts the N_k -channel mask $\mathbf{M} \in \mathbb{R}^{N_k \times H \times W}$ and \mathcal{F}_{ref} . Here, the target image \mathbf{I}_t is warped by the keypoint difference $\hat{\mathbf{k}} \in \mathbb{R}^{N_k \times 2}$, which is calculated as $\mathbf{k}_t - \mathbf{k}_s$. Accordingly, \mathbf{M} and \mathcal{F}_{ref} are obtained as follows:

$$\mathbf{M}, \mathcal{F}_{ref} = FE(\hat{\mathbf{H}} \oplus \mathcal{W}(\mathbf{I}_t, \hat{\mathbf{k}})). \quad (1)$$

\oplus and \mathcal{W} indicate concatenation and warping operation, respectively. $\mathcal{W}(\mathbf{I}, \alpha)$ means an image \mathbf{I} is warped by α . The warping operation is implemented using a bilinear sampler. FE consists of two 1×1 convolutional (Conv) blocks, five down blocks, and five up blocks [9].

Finally, we obtain the dense optical flow map \mathcal{F} as $\mathcal{F} = \sum_{i=1}^{N_k} \rho(\hat{\mathbf{k}}^i) \otimes \mathbf{M}^i + \mathcal{F}_{ref}$. Here, $\rho(\cdot)$ repeats the input tensor by $H \times W$ times and \otimes denotes element-wise multiplication. We adopt the Adam optimizer [14] with $\beta_1 = 0.5$, $\beta_2 = 0.999$ and the learning rate 0.0002.

* indicates equal contributions.

Also, we train the flow-based hair alignment for 240,000 iterations with batch size 8 in case of K-hairstyle dataset and 110,000 iterations with batch size 16 for VoxCeleb.

1.2 Hair Synthesis

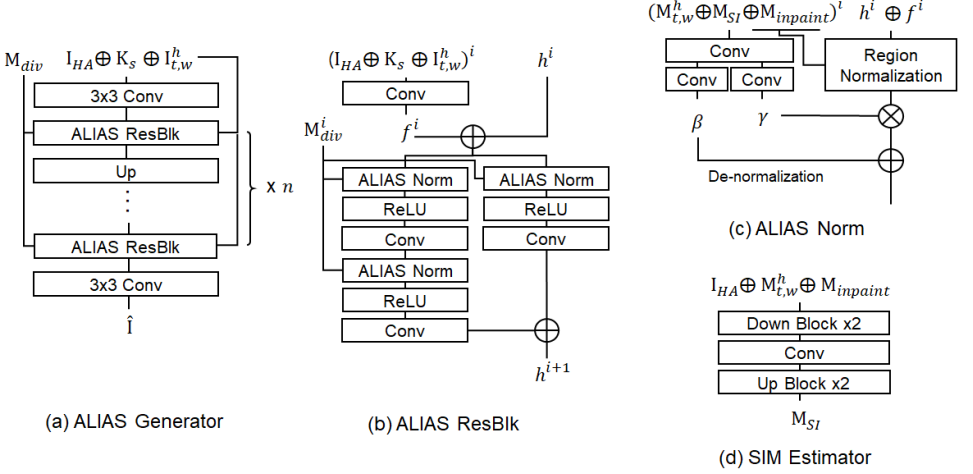


Figure 1: Detailed architecture of hair synthesis network.

The detailed architecture of the hair synthesis module is shown in Fig. 1. Our hair synthesis module synthesizes the aligned target hairstyle with the source image. The network needs to refine and apply the warped target hairstyle while preserving the source features such as the faces, clothes, or backgrounds. Furthermore, the network is also required to inpaint occluded regions in the source image with appropriate face, clothes, or backgrounds. To achieve this, we utilize ALIGNment-Aware Segmen (ALIAS) [40] generator with Semantic-region-aware Inpainting Mask (SIM) estimator.

ALIAS generator and discriminator. As described in Fig. 1 (a), the generator contains a series of ALIAS residual blocks (ResBlk), along with up-sampling layers. We use the multi-scale discriminator [40, 41].

ALIAS ResBlk. As presented in Fig. 1 (b), each ALIAS ResBlk consists of three sets of ALIAS normalization layer (ALIAS Norm), ReLU, and Conv layer. First, a resized hair-agnostic image $I_{HA}^i \in \mathbb{R}^{3 \times H^i \times W^i}$, an RGB-rendered facial keypoint image $K_s^i \in \mathbb{R}^{3 \times H^i \times W^i}$, and the warped target hair image $(I_{t,w}^h)^i$ are concatenated and fed to a Conv layer to obtain f^i . Then, f^i is concatenated with h^i , the feature from the previous layer, and injected to i -th ALIAS ResBlk.

ALIAS Norm. ALIAS Norm normalizes $h^i \oplus f^i$ separately based on the resized inpainting mask $M_{inpaint}^i \in \mathbb{R}^{1 \times H^i \times W^i}$. Then, the normalized features are de-normalized with affine parameters γ and β , estimated based on the resized M_{div}^i . M_{div}^i consists of three components, the warped target hair mask $M_{t,w}^{h,i}$, semantic-region-aware inpainting mask M_{SI}^i , and $M_{inpaint}^i$.

SIM estimator. The network separates $M_{inpaint}^i$ into face, clothes, background, and unknown region. SIM estimator consists of two Down blocks, one Conv layer, and two Up blocks. Each Down block has a Conv, Batch-Norm, and ReLU layer. Also, each Up block has an up-sampling layer, Conv, Batch-Norm, and ReLU layer. SIM estimator is trained end-to-end with ALIAS generator.

Losses. The details of the losses we use are described below.

\mathcal{L}_{cGAN} , \mathcal{L}_{FM} , and $\mathcal{L}_{percept}$. We adopt the conditional adversarial loss \mathcal{L}_{cGAN} , the feature matching loss \mathcal{L}_{FM} , and the perceptual loss $\mathcal{L}_{percept}$, referring to VITON-HD, SPADE, and pix2pixHD [4, 8, 10]. We use the hinge loss for \mathcal{L}_{cGAN} [10]. Let D be the discriminator and D^i be the activation of the i -th layer D . Similarly, VGG^i be the activation of the i -th layer VGG19 network [10]. N_{D^i} and N_{VGG^i} are the number of elements in D^i and VGG^i , respectively. Each of the above loss functions is described below.

$$\mathcal{L}_{cGAN} = \mathbb{E}[\log(D(\mathbf{M}_{div}, \mathbf{I}_s))] + \mathbb{E}[1 - \log(D(\mathbf{M}_{div}, \hat{\mathbf{I}}))] \quad (2)$$

$$\mathcal{L}_{FM} = \mathbb{E} \sum_{i=1}^T \frac{1}{N_{D^i}} \|D^i(\mathbf{M}_{div}, \mathbf{I}_s) - D^i(\mathbf{M}_{div}, \hat{\mathbf{I}})\|_1 \quad (3)$$

$$\mathcal{L}_{percept} = \mathbb{E} \sum_{i=1}^V \frac{1}{N_{VGG^i}} \|VGG^i(\mathbf{I}_s) - VGG^i(\hat{\mathbf{I}})\|_1 \quad (4)$$

Hairstyle loss \mathcal{L}_{style} . To capture the fine details of hairstyle features, we utilize the Gram matrix [9]. We compute the L2 distance between the gram matrices of the generated hair features and the target hair features extracted by VGG16 [10]. The generated hair features are obtained based on $\mathbf{M}_g^h \otimes \hat{\mathbf{I}}$ and the target hair features are obtained based on $\mathbf{M}_t^h \otimes \mathbf{I}_t$. G^i is the i -th Gram matrix, $G^i(v^i) = v^{i^T} v^i$, where $v^i \in \mathbb{R}^{H^i W^i \times N_{Ci}}$ is the activation of the i -th layer of VGG. Here, N_{Ci} and N_{Gi} represent the number of channels in VGG^i and in G^i , respectively. The activations from $\{relu1_2, relu2_2, relu3_3, relu4_3\}$ of VGG are used for the loss.

$$\mathcal{L}_{style} = \mathbb{E} \sum_{i=1}^V \frac{1}{N_{Gi}} \|G^i(VGG^i(\mathbf{M}_t^h \odot \mathbf{I}_t)) - G^i(VGG^i(\mathbf{M}_g^h \odot \hat{\mathbf{I}}))\|_2 \quad (5)$$

SIM estimator loss \mathcal{L}_{SIM} . \mathbf{GT}_{SIM} is a ground truth segmentation mask of the inpainting mask $\mathbf{M}_{inpaint}$. \mathbf{GT}_{SIM} is obtained from the source semantic masks of a face, clothes, and backgrounds extracted by the pre-trained face-parsing network [10]. We compute the binary cross-entropy loss between \mathbf{GT}_{SIM} and the predicted \mathbf{M}_{SI} as below.

$$\mathcal{L}_{SIM} = -\mathbb{E}[\mathbf{GT}_{SIM} \log(\mathbf{M}_{SI}) + (1 - \mathbf{GT}_{SIM}) \log(1 - \mathbf{M}_{SI})] \quad (6)$$

The total loss of the hair synthesis module is calculated as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{cGAN} + \lambda_{FM} \mathcal{L}_{FM} + \lambda_{percept} \mathcal{L}_{percept} + \lambda_{style} \mathcal{L}_{style} + \lambda_{SIM} \mathcal{L}_{SIM}, \quad (7)$$

where we set both λ_{FM} and $\lambda_{percept}$ to 10, λ_{style} to 50, and λ_{SIM} to 100.

We adopt the Adam optimizer [10] with $\beta_1 = 0$, $\beta_2 = 0.9$. The learning rate of the generator and the discriminator are set to 0.0001 and 0.0004, respectively. We train the hair synthesis module for 15,000 iterations with batch size 8 for both K-hairstyle and VoxCeleb.

2 Additional Qualitative Results

We conduct an additional qualitative comparison between our model and the baseline models. Fig. 2 and Fig. 3 present qualitative results of K-hairstyle dataset and VoxCeleb dataset, respectively. The results show that HairFIT successfully transfers hairstyles even when the source and the target image have different poses. Furthermore, our model preserves delicate target hairstyle features (e.g., curl, two-toned hair color, etc.) better than other models.



Figure 2: Qualitative comparison with K-hairstyle dataset. Due to the privacy issue, we blur the faces of the images.



Figure 3: Qualitative comparison with VoxCeleb dataset.

3 Limitations

Although HairFIT successfully transfers a hairstyle between images of different poses, our model has several limitations.

First, HairFIT has difficulty in aligning a target hair which has a significant occlusion of hair. Since our hair alignment module utilizes a warping operation to align the target hair, the module can rearrange the pixels of the existing hair but cannot newly generate the unseen hair. The first column of Fig. 4 presents an example where the right side of the target hair is extremely occluded. In this case, our model cannot transfer the right hair of the target image to the source image.

Next, a complicated texture or structure in occlusion regions degrades the quality of generated images. For instance, as described in the second column of Fig. 4, even though a person in a source image wears clothes with complex patterns, our model inpaints the region only with simple and general texture.

Lastly, HairFIT is dependent on hair segmentation masks. For example, if the target hair mask contains irrelevant regions such as the forehead, the output inevitably contains the region. On the third column of Fig. 4, the forehead of the output reflects the target forehead color which is different from the source since the target hair mask includes its forehead. Also, the last column of Fig. 4 indicates an example where the target hair mask does not contain thin hair on the forehead, leading to an inaccurate hairstyle transfer.



Figure 4: Limitations of HairFIT. The first column describes the case where a target hair has an extreme occlusion and the second column is an example of complicated occlusion regions in a source image. The last two columns are the cases of limitation due to inaccurate hair segmentation masks.

4 Comparison with StarGAN v2

The previous work [10] shows that StarGAN v2 [10] has a capability to modify the hairstyle of the source image based on the given target image. However, we found that StarGAN v2 fails to preserve other features (*e.g.*, skin color, face shape, clothes, etc.) of the source image, which are essential to maintain the source person’s identity. Since StarGAN v2 changes both the hairstyle and the identity of the source image, it is not appropriate to compare StarGAN

Dataset	K-hairstyle	VoxCeleb
StarGAN v2 (Gender)	0.5238	0.9938
StarGAN v2 (Hair)	0.5590	-
HairFIT (Ours)	0.9892	0.9993

Table 1: Quantitative comparison of identity preservation performance between StarGAN v2 and HairFIT. We measure the face verification accuracy using the pretrained ArcFace with K-hairstyle and VoxCeleb dataset.

v2 with our model. For this reason, none of the existing hairstyle transfer work such as MichiGAN, LOHO considered StarGAN v2 as their baseline, either.

To evaluate the identity preservation performance, we measure the face verification accuracy of StarGAN v2 compared to HairFIT using the pre-trained ArcFace [9], which is one of the state-of-the-art face recognition models. Since StarGAN v2 requires the domain labels for training, we utilize the gender labels and the hairstyle labels provided from K-hairstyle dataset and the gender labels from VoxCeleb dataset. As shown in Table 1, HairFIT successfully preserves the source identity in both K-hairstyle and VoxCeleb. On the other hand, StarGAN v2 trained with K-hairstyle fails to preserve the source identity. Although StarGAN v2 trained with VoxCeleb achieves the high verification accuracy, Fig. 5 demonstrates that StarGAN v2 also modifies the features (*e.g.*, skin color, makeup style, etc.) related to the source identity. This reason makes StarGAN v2 hardly applicable for hairstyle transfer.



Figure 5: Qualitative comparison of identity preservation performance between StarGAN v2 and HairFIT.

References

- [1] Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2021.
- [2] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2020.
- [3] Jiankang Deng, Jia Guo, Xue Niannan, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019.
- [4] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.
- [5] Xun Huang and Serge J. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. of the IEEE international conference on computer vision (ICCV)*, 2017.
- [6] Taewoo Kim, Chaeyeon Chung, Sunghyun Park, Gyojung Gu, Keonmin Nam, Wonzo Choe, Jaesung Lee, and Jaegul Choo. K-hairstyle: A large-scale korean hairstyle dataset for virtual hair editing and hairstyle classification. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, 2021.
- [7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. the International Conference on Learning Representations (ICLR)*, 2015.
- [8] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019.
- [9] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019.
- [10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. the International Conference on Learning Representations (ICLR)*, 2015.
- [11] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018.
- [12] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2018.
- [13] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018.