PairedCycleGAN: Asymmetric Style Transfer for Applying and Removing Makeup

Huiwen Chang
Princeton University

Jingwan Lu
Adobe Research

Fisher Yu
UC Berkeley

Adam Finkelstein
Princeton University

Abstract

This paper introduces an automatic method for editing a portrait photo so that the subject appears to be wearing makeup in the style of another person in a reference photo. Our unsupervised learning approach relies on a new framework of cycle-consistent generative adversarial networks. Different from the image domain transfer problem, our style transfer problem involves two asymmetric functions: a forward function encodes example-based style transfer, whereas a backward function removes the style. We construct two coupled networks to implement these functions – one that transfers makeup style and a second that can remove makeup – such that the output of their successive application to an input photo will match the input. The learned style network can then quickly apply an arbitrary makeup style to an arbitrary photo. We demonstrate the effectiveness on a broad range of portraits and styles.

1. Introduction

Digital photo manipulation now plays a central role in portraiture. Professional tools allow photographers to adjust lighting, remove blemishes, or move wisps of hair. Whimsical applications let novices add cartoon elements like a party hat or clown nose, or to turn photos into drawings and paintings. Some tools like Taaz [2] and PortraitPro [1] can digitally add makeup to a person in a photo, but the styles are limited to a collection of preset configurations and/or a set of parameters that adjust specific features like lip color.

This paper introduces a way to digitally add makeup to a photo of a person, where the style of the makeup is provided in an example photo of a different person (Figure 1). One challenge is that it is difficult to acquire a dataset of photo triplets from which to learn: the source photo, the reference makeup photo, and the ground truth output (which preserves identity of the source and style of the reference). Previous work on style transfer avoids the need for such a training set by defining the style and content loss functions based on deep features trained by neural networks [8, 16, 18]. While those approaches can produce good results for stylization of imagery in general, they do not work well for adding various makeup styles to faces. A second challenge, specific to our makeup problem, is that people are highly sensitive to visual artifacts in rendered faces. A potential solution is to restrict the stylization range so as to define a specific color transformation space (such as affine transformations), or so as to preserve edges [18, 19, 16]. Unfortunately, this approach limits the range of makeup, because many styles include features that would violate the edge preservation property such as elongated eyelashes or dark eye liner.

Inspired by recent successful photorealistic style transfer based on generative adversarial networks (GANs), we take an unsupervised learning approach that builds on the CycleGAN architecture of Zhu *et al*. [26]. CycleGAN can transfer images between two domains by training on two sets of images, one from each domain. For our applica-

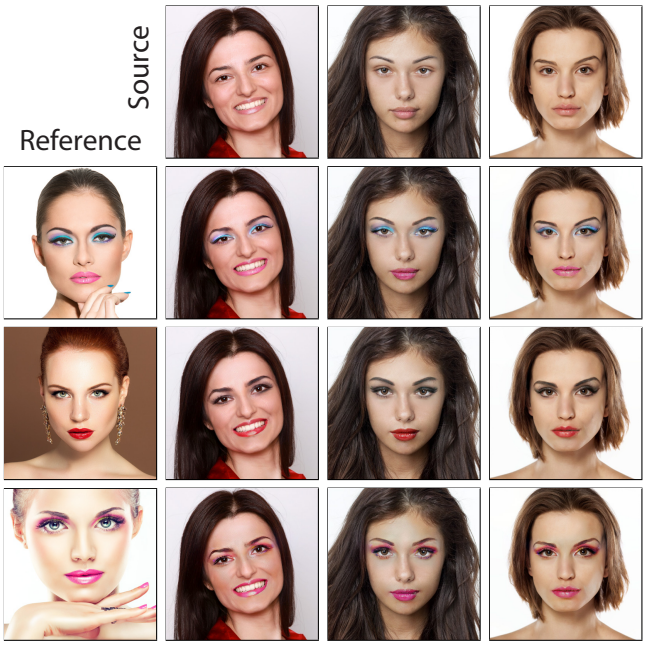


Figure 1: Three source photos (top row) are each modified to match makeup styles in three reference photos (left column) to produce nine different outputs (3×3 lower right).

tion, CycleGAN could in principle learn to apply a general make-you-look-good makeup to a no-makeup face, but it would not replicate a specific example makeup style.

Thus, we introduce a set of problems where the forward and backward functions are asymmetric. Another example application would be transferring patterns from an example shirt to a white shirt, where the paired backward function could remove patterns from a shirt to make it white. Such forward (style transfer) functions require a source image and reference style as input, whereas the backward function (style removal) only takes the stylized image as input.

Our approach relies on two asymmetric networks: one that transfers makeup style and another that removes makeup (each of which is jointly trained with an adversary). Application of both networks consecutively should preserve identity of the source photo (Figure 2). Finally, to encourage faithful reproduction of the reference makeup style, we train a style discriminator using positive examples that are created by warping the reference style face into the shape of the face in the source photo. This strategy addresses the aforementioned problem of a ground truth triplet dataset.

The principal contributions of this paper are: (1) A feed-forward makeup transformation network that can quickly transfer the style from an arbitrary reference makeup photo to an arbitrary source photo. (2) An asymmetric makeup transfer framework wherein we train a makeup removal network jointly with the transfer network to preserve the identity, augmented by a style discriminator. (3) A new dataset of high quality before- and after-makeup images gathered from YouTube videos.

2. Related Work

Makeup Transfer and Removal. Makeup transfer is a specific form of style transfer that demands precise semantic understanding of the face to generate photorealistic details. Several previous work addressed the challenges of makeup transfer. Tong *et al*. [23] tackled the problem of automatic transfer of makeup from a makeup reference to a new portrait. Similar to image analogies [10], their framework requires as reference a pair of well-aligned before-makeup and after-makeup photos of the same person. Guo *et al*. [8] proposed a method that requires only the after-makeup photo as reference. They decompose the reference and target portrait into three layers, face structure, skin detail and color, and transfer the makeup information for each layer separately. A major disadvantage of the approach is the need of a pre-processing step to warp the example makeup to the target face based on detected facial landmarks. Similarly, Li *et al*. [15] proposed to decompose the source portrait into albedo, diffuse and specular layers and transform each layer to match the optical properties of the corresponding layers of the reference. Different from previous work that transfer makeup from one reference, Khan *et al*. [14] introduced an approach to transfer local makeup styles of individual facial components from multiple makeup references. corresponding facial components in the target. Inspired by the recent success of neural-based style transfer techniques, Liu *et al*. [18] applied optimization-based neural style transfer models locally on facial components.

Other than makeup transfer, researchers have also attempted to digitally remove makeup from portraits [24, 4]. All previous work treat makeup transfer and removal as separate problems. We propose a single framework that can perform both tasks at the same time and we show that by alternating the improvement of transfer and removal processes, better results can be obtained for both tasks.

General Style Transfer. Researchers have investigated the general style transfer problems extensively. Gatys *et al*. [6] studied artistic style transfer. They proposed to combine the content of one image with the style of another by matching the Gram matrix statistics of deep features using optimization. Johnson *et al*. [13] later proposed a feed-forward network to approximate the solution to the optimization problem when the goal is to transfer a single style to arbitrary target images. The above methods are designed for painterly or artistic style transfer. When both the target and the style reference are photographs, the output exhibits artifacts reminiscent of a painterly distortion. In order to transfer photographic style, recent work [19] added semantic segmentation as an optional guidance and imposed a photorealism constraint on the transformation function to avoid edge distortions in the result.

Style transfer can also be considered as image analogy with a weak supervision, as described in [16, 9]. Liao *et al*. assume the input image pairs have similar semantic structure and use PatchMatch to calculate the dense correspondences in deep feature space. Their approach works well for both artistic and photorealistic style transfer, but is computationally expensive. The follow-on work by He *et al*. [9] improves the formulation for color transfer tasks. Both approaches showed good makeup transfer results by treating it as a color transfer problem. However, we argue that makeup transfer is more than color transfer. Sophisticated eye makeups require the generation of new edges to imitate the example eye lashes. Lip makeup alters not only the color, but also the textural appearance, e.g. shininess, of the lip. We resort to feed-forward neural networks to learn these complicated transfer behaviors from data.

Image Domain Adaption. Style transfer can also be posed as a domain adaptation problem in the image space. Image domain adaptation methods learn a mapping function to transform images in a source domain to have similar appearance to images in a target domain. Taigman *et al*. [22] first formulated the problem and adopted generative adversarial network (GAN) [7] and variational autoencoder

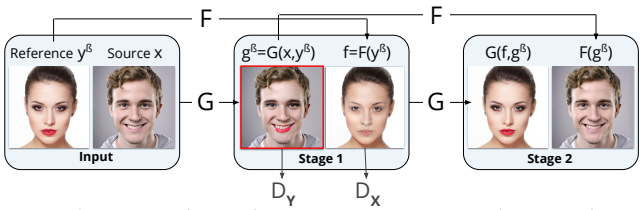


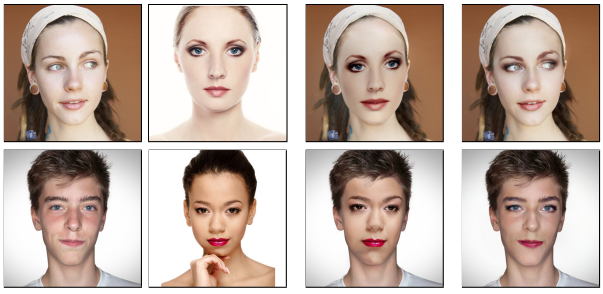
Figure 2: Network Pipeline. Given a source photo (without makeup) and a makeup reference, our system simultaneously learns a makeup transfer function G and a makeup removal function F. The result of the first stage can be viewed as a pair output by image analogy, and can serve as input for the second stage. We compare the output from the second stage with the source to measure identity preservation and style consistency.

(VAE) [20] as the mapping function to enforce the transformed output to be similar to the source in feature space. Zhu *et al*. [26] introduced CycleGAN which uses generative network together with a cycle consistency loss to encourage the distribution of the mapped images to be indistinguishable from that of the real images in the target domain. Similarly, Liu *et al*. [17] employed GAN and VAE and proposed a shared latent space assumption which is shown to imply the cycle-consistency constraints in CycleGAN.

To use domain adaptation such as CycleGAN for style transfer, the training procedure requires a set of style images in the *same* style and a set of target images of similar content. The learned mapping function takes one target image as input and transforms it into the style domain. For the makeup application, CycleGAN would either need a set of faces with the same makeup to train a network for each makeup style; otherwise it could only learn to apply a general make-you-look-good makeup to a no-makeup face. Instead, we formulate forward and backward mapping as asymmetric functions, and introduce variants of cycle consistency losses to ensure the successful transfer of color, structure, and high frequency details from a particular makeup example.

3. Formulation

Due to the lack of pixel-aligned before-makeup and after-makeup image pairs, we tackle the makeup transfer problem using an unsupervised approach. Let X and Y be the no-makeup and with-makeup image domains where no pairings exist between the two domains. To characterize the no-makeup image domain X, we use a collection of no-makeup portrait photos of diverse facial features, expressions, skin tones, and genders, $\{x_i\}_{i=1,...,N}, x_i \in X$. To model the with-makeup image domain Y, we use a diverse set of makeup examples $\{y_j\}_{j=1,...,M}, y_j \in Y$, where the makeup styles range from nude and natural to artistic and intense. $Y^\beta \subset Y$ refers to a sub-domain of Y that contains images of a specific makeup style β. When $y_j \in Y^\beta$, we denote it as y_j^β.



Source Reference Warping results Our results

Figure 3: Warping Guidance. We extract face landmarks and warp the reference face to source landmarks. The warping results exhibit not only the style but also pose, lighting, and some identity properties from the reference. The resulting face looks like a mix between the source and reference in terms of identity. We use the reference and its warping result as the positive examples of our discriminator L_S. In comparison, our results match the style of reference and the identity of source better.

As illustrated in Figure 2, our key idea is to simultaneously train two separate neural networks G and F, one to transfer specific makeup style and another to remove makeup. We hope by using diverse training examples, the network G can generalize its learning to the entire with-makeup image domain and can transfer arbitrary makeup styles to arbitrary faces at run time. Given a photo of a person with makeup, $y^\beta \in Y^\beta$, and a photo of a different person without makeup, $x \in X$, the makeup transfer network $G : X \times Y^\beta \to Y^\beta$ extracts the makeup layer from y^β and applies it to x maintaining its identity. Our result $G(x, y^\beta)$, highlighted in Figure 2, should belong to the domain Y^β. Given the same photo y^β, the demakeup network $F : Y \to X$ learns to remove the makeup maintaining the identity of y^β. Note that G and F are unbalanced functions. G takes a pair of images as input to transfer the style from one to the other. F removes makeup given the with-makeup image itself. If G and F are successful, the output of G can be used as a makeup example to be transferred to the output of F, doubling the number of training examples. Also, if G and F operate without changing the facial features, transferring the makeup style from person 1 to 2 and then back to 1 should generate exactly the two input images. Assume x is sampled from X i.i.d according to some distribution $\mathcal{P}_X$ and y^β is sampled from Y i.i.d according to some distribution $\mathcal{P}_Y$. Based on makeup transfer properties, we adopt the following losses.

Adversarial loss for G. We first employ an adversarial loss to constrain the results of G to look similar to makeup faces from domain Y. The loss is defined as:

$$L_G(G, D_Y) = \mathbb{E}_{y\sim\mathcal{P}_Y}[logD_Y(y)] + \mathbb{E}_{x\sim\mathcal{P}_X, y\sim\mathcal{P}_Y}[log(1 - D_Y(G(x, y)))] \quad (1)$$

where the discriminator D_Y tries to discriminate between

the real samples from domain Y and the generated samples $G(x, y^\beta)$, and the generator G aims to generate images that cannot be distinguished by the adversary D_Y.

Adversarial loss for F. We also apply an adversarial loss to encourage F to generate images indistinguishable from the no-makeup faces sampled from domain X.

$$L_F(F, D_X) = \mathbb{E}_{x\sim\mathcal{P}_X}[logD_X(x)] + \mathbb{E}_{y^\beta\sim\mathcal{P}_Y}[log(1 - D_X(F(y^\beta)))] \quad (2)$$

Identity loss. The adversarial loss constrains the output of G to look like a face with makeup applied. A trivial mapping function $G(x, y^\beta) = y^\beta$, i.e. G generates a result identical to y^β, will satisfy the above constraint, but is not a desirable solution due to the loss of identity of x. In fact, preserving the identity in the makeup transfer process is a challenging problem. Previous work [16] needed to apply additional post-processing step to recover the lost identity sacrificing the fidelity of the transfer. We propose to use the demakeup function F to explicitly encourage the preserving of identity in the makeup transfer process. The idea is that if we apply makeup to x and then immediately remove it, we should get back the input image x exactly. We use L1 loss to penalize the difference between $F(G(x, y^\beta))$ and x:

$$L_I(G, F) = \mathbb{E}_{x\sim\mathcal{P}_X, y^\beta\sim\mathcal{P}_Y}[||F(G(x, y^\beta)) - x||_1] \quad (3)$$

Style loss. The previous losses constrain the output of G to lie on the manifold of Y (faces with makeup) while maintaining the identity of X. However, they are not sufficient to ensure the successful transfer of details of a particular makeup style y^β. For this purpose, we propose two style losses, L1 reconstruction loss L_S and style discriminator loss L_P.

One key observation is that if we transfer the makeup style from face y^β to face x, and then use the result $G(x, y^\beta)$ to transfer the same makeup style back to the makeup-removed face $F(y^\beta)$, the result $G(F(y^\beta), G(x, y^\beta))$ should look exactly like the input face with makeup y^β:

$$L_S(G, F) = \mathbb{E}_{x\sim\mathcal{P}_X, y^\beta\sim\mathcal{P}_Y}[||G(F(y^\beta), G(x, y^\beta)) - y^\beta||_1] \quad (4)$$

Using L1 loss in the pixel domain helps the transfer of general structure and color (e.g. the shape of eyebrows and the gradient of eye-shadow), but leads to blurry results incapable of transferring sharp edges (e.g. eyelashes and eyeliners). Therefore, we add an auxiliary discriminator D_S, which decides whether a given pair of faces wear the same makeup. During training, we need to feed D_S with real makeup pairs (y^β, the same makeup style β applied to another face) and fake makeup pairs (y^β, $G(x, y^\beta)$). The challenge is that for the real makeup pairs, we do not have the *ground-truth* of applying makeup style β to another face, since all makeup styles appear only once in our training set. Therefore, we generate a *synthetic ground-truth* $W(x, y^\beta)$, by warping the reference makeup face y^β to match the detected facial landmarks in the source face x. Figure 3 show two example warping results. Subtle facial details important for preserving identity and expression are lost. For example, in the top row, the nose structure is changed and the laugh lines are completely removed. In the bottom row, the facial hair disappeared and the skin tone is different between the face and the neck regions. Though the warped images cannot serve as the final results, they can offer the discriminator a clue about what should be classified as positive examples of different faces wearing the same makeup. The loss for D_S is defined as:

$$L_P(G, D_S) = \mathbb{E}_{x\sim\mathcal{P}_X, y^\beta\sim\mathcal{P}_Y}[logD_S(y^\beta, W(x, y^\beta))] + \mathbb{E}_{x\sim\mathcal{P}_X, y^\beta\sim\mathcal{P}_Y}[log(1 - D_S(y^\beta, G(x, y^\beta)))] \quad (5)$$

Total Loss. We optimize a min-max objective function $\min_{G,F}\max_{D_X,D_Y,D_S} L$, where the loss L is defined as:

$$L = \lambda_G L_G + \lambda_F L_F + L_I + L_S + \lambda_P L_P \quad (6)$$

λ_G, λ_F, and λ_P are the weights to balance the multiple objectives. The next section will provide more training details and discuss the appropriate weights.

4. Implementation

Training Pipeline. We aim to generate high-resolution makeup transfer results, but existing generative networks can only produce images smaller than 512×512 due to limited GPU memory. Motivated by the observation that the makeup of eye regions differs a lot from that of the skin or lip regions, we train three generators separately focusing the network capacity and resolution on the unique characteristics of each individual regions. Given each pair of no-makeup and with-makeup images, we first apply face parsing algorithm to segment out each facial component, e.g. eyes, eyebrows, lip, nose, etc. (Figure 4). To best transfer eye makeup, we calculate a circle enclosing one eye, the corresponding eyebrow and the skin pixels

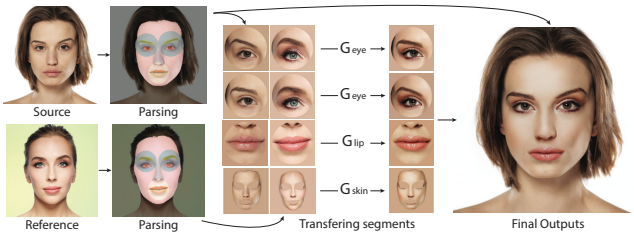


Figure 4: Generator per Segment. For each image, we apply face parsing algorithm to segment out each facial component. And we train three generators and discriminators separately for eyes, lip and skin considering the unique characteristics of each regions.

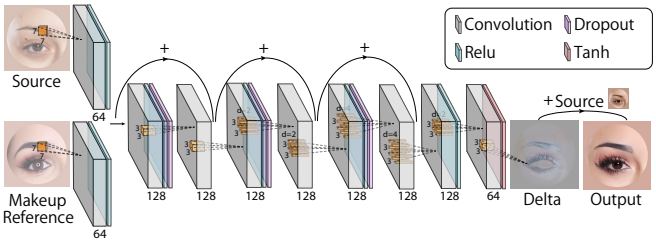


Figure 5: Generator Architecture. For generator G, we use DRN [25] with 3 dilated residual blocks to retain small spatial information (such as eye makeup). d indicates the dilation factor. We use two degridding layers (a 2-dilated 3×3 convolution and a 3×3 convolution) at the end of the network to avoid grid artifacts. The architecture of F is similar. The only difference is that it takes one image as input and therefore does not need concatenation.

in between and we train the generator on the entire eye region. Note that we flip the right eye regions horizontally so that we only need to train a single network for the left eye regions. We use circles for the eye and lip cutout due to the simplicity of applying random rotations as data augmentation. For skin and lip, we also perform horizontal flipping to double the amount of training data. As post-processing, we blend the generated pixels of the eye, lip and skin regions into the source using Poisson blending.

Data Collection. Existing face datasets are collected for face recognition and identification purposes and are of low image resolution (below 640x480). Faces without makeup and with makeup are often mixed together. For our unsupervised learning problem, we need two separate high-resolution datasets, one containing faces without any makeup and another one containing faces with a large variety of makeup styles. To this end, we collect our own datasets from Youtube makeup tutorial videos. For each video, we extract frames from the first quarter and the last quarter since they likely include no-makeup face and after-makeup face. We discard the middle section of the video since the intermediate frames are most likely portraying the on-going makeup process. Among the extracted frames, we discard the blurry and duplicate ones and the ones containing face regions smaller than 400x400. We classify the remaining frames as either no-makeup or with-makeup using a heuristic algorithm which detects whether the eye regions are coated with non-skin colors or rich textures. After that, we ask the Amazon Mechanical Turk (MTurk) users to validate whether each frame is indeed a sharp no-makeup or with-makeup face with eyes open and without occlusions by fingers, brushes, etc. In this way, we harvest a no-makeup dataset of 1148 images and a with-makeup dataset of 1044 images. Our datasets contain a wide variety of facial identities and makeup styles.

Network Architecture. A reasonable architecture choice for the generators G and F is the traditional encoder-decoder network [21], which progressively downsamples the input image encoding it into a compact hidden code and then progressively upsamples the hidden code to reconstruct an image of the same resolution to the input. As discussed in pix2pix [11], a network architecture requiring the information to flow through a low-resolution bottleneck layer is not capable of generating sharp high-frequency details. To circumvent the information loss, they added skip connections, following the general shape of a U-Net [21]. U-net can retain the prominent edges in the input image and successfully hallucinate new edges, such as eyelashes; but it is not able to handle scale and orientation differences, such as transferring makeup from smaller eyes to larger eyes. We also considered adopting spatial transformation network (STN) [12]. STN can generate new edges by transforming the reference makeup, but it suffers from the same problem as warping (Figure 3) that the identity of the source image is often lost due to the direct copy and paste of pixels from the reference. Instead of U-net and STN, we use Dilated ResNet (DRN) [25] architecture for the generators. Dilated convolution increases the receptive fields of the deeper network layers while maintaining the spatial resolution to avoid information loss. DRN utilizes the high-resolution feature maps to preserve image details. Its degridding layers can further improve the accuracy of spatial predictions. Our generator network contains 3 dilated residual blocks as shown in Figure 5. We also add 2 degridding layers at the end of the network to avoid artifacts in the generated results. Our generator G takes two images as input, as plotted in Figure 5, while our generator F only takes one with-makeup image as input and hence needs no concatenation.

Instead of directly generating the output pixels, the generator G calculates the delta image which can be added to the source image to obtain the final output, as illustrated in Figure 5. By doing that, we hope to maintain the original skin tone and lighting environment in the source and only transfer the makeup as an add-on layer. The skin on the face can be smoothed and sculpted by contours and highlights, but the general skin color should be similar to the neck for natural results. In this regard, we encourage the delta image to be sparse, and add a regularization term $L_R = ||G(x, y^\beta) - x||_1$ with weight 0.1 to our objective function. Our discriminators follow the design of the 256×256 discriminator in pix2pix [11]. Since faces contain distinctive global structures, we have the discriminator look at the whole image instead of image patches.

Training Details. We pretrain F using CycleGAN [26]. Makeup transfer is a one-to-many transformation and makeup removal is many-to-one. CycleGAN can remove most of the makeup from a face, but cannot transfer a specific makeup style to a new face. With F initialized by CycleGAN, we alternate the training of G and the fine-tuning of F. Since G is a much harder problem and F gets a good head start, we train G ten times more frequently than F.

Figure 6: Network Architecture and Loss Analysis. We compare our network using DRN architecture with losses described in Section 3, with models trained without specific loss terms and with the model trained using U-net architecture.

For the first 200 epochs, we set $\lambda_G = \lambda_F = \lambda_P = 0.1$ and after that, we trust the discriminators more and raise these values: $\lambda_G = \lambda_F = \lambda_P = 0.5$. The lip and face networks are trained for 400 epochs while eyes are trained for 850 epochs.

5. Results

In Figure 6, we first analyze whether each loss term is necessary by training a separate generator each time with one loss removed from our energy function. We keep all the hyper parameters the same as described before. When we remove L_G, the GAN loss for generator, G could apply the eye shadow anywhere around the eye on the source image since there is no discriminator distinguishing whether it is realistic or not. When we remove L_I, the generator encodes the characteristics of eyes in both source and reference resulting in identity loss. Without L_S, the results look less saturated and the makeup style is not fully transferred. Without L_P, the color in the result become more vivid, but some eye lashes get neglected. We also tried the U-net architecture with 9 blocks as described in the work[11]. But for our problem, DRN performs better than U-net architecture.

Figure 7 shows results on the lip and eye regions. Our network can faithfully reproduce the makeup material properties in the reference. In the top two rows, the generated lips not only exhibit plausible colors, but also inherit the shiny appearance from the reference. Our network can also synthesize high frequency eye lashes reasonably well. We would like to point out that we do not perform any pre-alignment on the facial features. With random rotation as data augmentation, our network is able to learn rotation, scaling and other transformations inherently and put makeup components in the right places. Notice that when the distance between eyes and eyebrows or the orientation of eyebrows are very different between the source and the reference, we can still synthesize plausible results (bottom two rows).

Figure 8 shows the combined results. Our network can transfer a variety of makeup styles across people of different skin tones preserving the original skin tone and other important identity details in the source.

However, one limitation of our approach is that the network does not work as well on extreme makeup styles unseen during training. As shown in Figure 9, the eye makeup is very different from the majority of the training examples. Our synthesized eyes look plausible but lack the precise pointy wing from the reference. The reference makeup contains shiny sparkles on the face which is unseen during training and is therefore not transferred in the results.

Quantitative Comparison We conducted a user study on Amazon Mechanical Turk making a pairwise comparison among results of the method of Liao et al. [16], of Liu et al. [18], and of our method. We randomly select 102 source photos, and assign 3 of them to each of 34 style images, so we have 102 groups of source-style inputs. We then pick a pair of results from the same group to compare, and we ask 10 or more subjects to select which result better matches the makeup style in the reference. On

Figure 7: Results of the lip and eye regions.

Figure 8: Results on the entire face.

average 58.2% of people prefer our results over those of Liu et al., and 78.9% of people prefer ours over those of Liao et al.. Note that sometimes the results of Liu et al. may not look realistic, but this study focuses only on style similarity. Given the result images in the same source-makeup group, we employed the Bradley Terry model [3] to derive a quantitative performance score. Figure 10 shows ranking statistics – a count of how often a method had each rank. Our method outperforms all others by this measure.

Makeup Transfer Comparison. In Figure 11, we compare our results with three different previous work, our implementation of *makeup like a super star* [18], *example-based portrait stylization* [5] and *deep image analogy* [16]. We cannot compare with the work by Tong *et al*. [23] or Khan *et al*. [14], because they do not solve the same problem as ours and require more inputs, e.g. before-makeup and after-makeup pair or multiple makeup references. For *makeup like a super star* [18], accurate face segmentation is crucial. We manually refined our face parsing result and included an additional eye shadow label. Their main problem is that different head poses and lighting conditions may lead to asymmetric eye makeup in the results. The *portrait stylization* work [5] focuses on transferring artistic and painterly style, and sometimes distorts facial features that are only visible when transferring photographic style. We apply their method in the face region only and alpha composite the result onto the rest of the image. Similarly, we apply *deep image analogy* [16] in the face region. It finds dense correspondences in feature space between the source and reference. When the expressions differ (mouth open versus mouth closed), or the makeup style is dramatic (bottom row), the correspondences cease to exist and hence the

Figure 9: Limitations. Extreme makeup styles (dark wings, face sparkles) unseen during training are difficult to reproduce.

analogy results are not as good. In favor of more realistic results and less distortions, they adopt a post-processing refinement step for photo-to-photo analogy, which transfers colors from the reference makeup and retains the structures in the source. The refinement step helps to preserve the identity but harms the system's capability to introduce high frequency details. Previous work also tend to alter the skin tone of the source with the skin tone in the reference resulting in identity loss that deviates from the goal of most professional makeup artists. For example, the results by Liao *et al*. contain unnatural skin tone transition from the neck to the face. In contrast, our network takes as input makeup reference of arbitrary head pose and facial expression, and is capable of properly transferring makeup styles, from natural to dramatic, preserving the source identity. Our network also maintains the input skin tone as much as possible by distilling only the makeup layer from the reference for transfer. We include more qualitative and quantitative results in the supplementary material.

Makeup Removal Comparison. Restoring the natural look behind makeup is an ill-posed problem. There could be many plausible faces behind the same makeup, for example, the blemishes could be covered by the foundation and a natural pink lip could be painted red. Our generator tries to offer a plausible prediction of one's natural face given the makeup face as input. It may be beneficial for face verification systems. We compare our makeup removal results with "face behind makeup" [24] in Figure 12. Both methods produce natural-looking before-makeup face. But our network removes makeup more aggressively and generates sharper results. The results by Wang *et al*. [24] retain partial eye-shadows, eyeliners and eyebrow colors. On the contrary, we remove them all and recover natural under-eye bags and suggest tentative appearances for the original

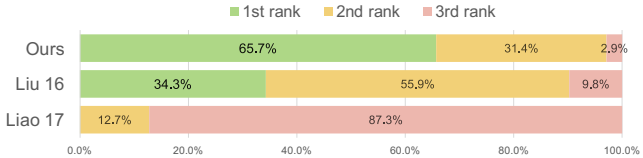


Figure 10: In paired comparison, how well do various methods perform, as judged by subjects on MTurk?

Figure 11: Makeup Transfer Results. We compare with makeup and portrait style transfer work [18, 5, 16].

eyebrows and lips. Our network also better preserves the original skin tone while removing highlights, contours and blushes. One limitation is that we cannot reconstruct realistic blemishes since most of the no-makeup faces in our training set contain clear and smooth skin.

6. Conclusion

We present an unsupervised learning approach for transferring arbitrary makeup styles to arbitrary source faces and for removing makeup, both at interactive rates. We introduce the idea of asymmetric style transfer and a framework for training both the makeup transfer and removal networks together, each one strengthening the other. Compared with previous work, our system generates more convincing results more quickly, and significantly improves the preservation of facial identity in the source photo. We believe this novel unsupervised learning framework can be used in other domains ranging from similar applications like automatic aging and de-aging, to further afield, like photorealistic object style transfer.

Figure 12: Demakup Results. We compare with makeup removal work by Wang et al. *[24]. Our demakeup network F can remove the detected makeup to virtually recover the original face.*

References

[1] Portrait pro - easy photo editing software. http://www.portraitprofessional.com/. 1

[2] Taaz virtual makeover and hairstyles. http://www.taaz.com/. 1

[3] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. 7

[4] Y.-C. Chen, X. Shen, and J. Jia. Makeup-go: Blind reversion of portrait edit. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[5] J. Fišer, O. Jamriška, D. Simons, E. Shechtman, J. Lu, P. Asente, M. Lukáč, and D. Sỳkora. Example-based synthesis of stylized facial animations. *ACM Transactions on Graphics (TOG)*, 36(4):155, 2017. 7, 8

[6] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 2

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[8] D. Guo and T. Sim. Digital face makeup by example. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 73–79. IEEE, 2009. 1, 2

[9] M. He, J. Liao, L. Yuan, and P. V. Sander. Neural color transfer between images. *CoRR*, abs/1710.00756, 2017. 2

[10] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340. ACM, 2001. 2

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016. 5, 6

[12] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *CoRR*, abs/1506.02025, 2015. 5

[13] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. 2

[14] A. Khan, Y. Guo, and L. Liu. Digital makeup from internet images. *CoRR*, abs/1610.04861, 2016. 2, 7

[15] C. Li, K. Zhou, and S. Lin. Simulating makeup through physics-based manipulation of intrinsic image layers. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4621–4629, 2015. 2

[16] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. Visual attribute transfer through deep image analogy. *arXiv preprint arXiv:1705.01088*, 2017. 1, 2, 4, 6, 7, 8

[17] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. *arXiv preprint arXiv:1703.00848*, 2017. 3

[18] S. Liu, X. Ou, R. Qian, W. Wang, and X. Cao. Makeup like a superstar: Deep localized makeup transfer network. *arXiv preprint arXiv:1604.07102*, 2016. 1, 2, 6, 7, 8

[19] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. *arXiv preprint arXiv:1703.07511*, 2017. 1, 2

[20] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 3

[21] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. 5

[22] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. 2

[23] W.-S. Tong, C.-K. Tang, M. S. Brown, and Y.-Q. Xu. Example-based cosmetic transfer. In *Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on*, pages 211–218. IEEE, 2007. 2, 7

[24] S. Wang and Y. Fu. Face behind makeup. *AAAI Conference on Artificial Intelligence*, 2016. 2, 7, 8

[25] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[26] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017. 1, 3, 5