

DIGITAL IMAGE PROCESSING

FINAL PROJECT TEAM:25

¹ *Sakura Yamaki (八卷櫻)*,² *Chun Ting Wu (吳均庭)*,

¹ Department of Computer Science and Information Engineering,
National Taiwan, Taipei, Taiwan,
E-mail: r07922153@ntu.edu.tw r06922055@ntu.edu.tw

ABSTRACT

Facial makeup transfer aims to translate the makeup style from a given reference makeup face image to another non-makeup one while preserving face identity. Such an instance-level transfer problem is more challenging than conventional domain-level transfer tasks, especially when paired data is unavailable. Makeup style is also different from global styles (e.g., paintings) in that it consists of several local styles/cosmetics, including eye shadow, lipstick, foundation, and so on. Extracting and transferring such local and delicate makeup information is infeasible for existing style transfer methods. So we use the issue by incorporating both global domain-level loss and local instance-level loss in a dual input/output Generative Adversarial Network, called BeautyGAN[2]. BeautyGAN generates and outputs an image close to the makeup style of the image which show make-up style on the video image which is capture from web camera. We upload the source code in github (https://github.com/sakurayamaki/Generating-makeup-image-with-BeautyGAN-using-web-camra/tree/master/source_code). Please try it and enjoy by them.

Keywords: Facial makeup transfer, *GAN*, *Live streaming*.

1. INTRODUCTION

Live broadcasting is one of the most popular industries recently. People use live video to share their lives and some even make money from the fast growing market and also many paper concern to make-up transfer system such as [1, 2, 3, 35] are published recently. During broadcasting, face filters are often used to make you look more stylish and appealing in the live video. However, existing services only provide few pre-defined styles and might not be suitable for anyone. In this project, the ultimate goal is to make a flexible face filtering system that allows users to customize their styles by providing images or their favorite styles, and our system is able to extract the styles, such as, make-up, skin tone, and so on, and apply them to the user faces.

2. PROBLEM DEFINITION

Input a target and example face, we would like to develop a model that extracts the style from the example and output the target face with the example style applied.

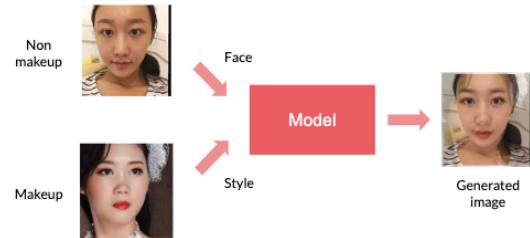


Figure 1: Outline of our method.

3. EXPECTED RESULTS

Input a sequence of video and an image of face, and we want to extract the style information from the image and apply it to the video.

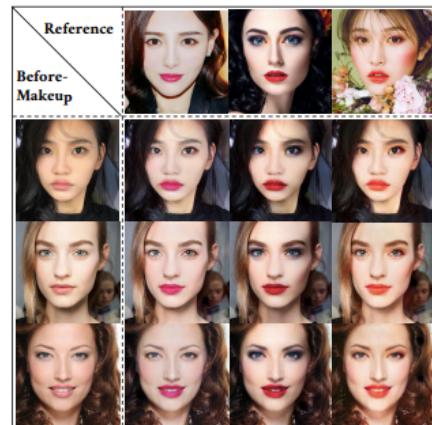


Figure 2: Example results of BeautyGAN model[2] for makeup transfer. Three makeup styles on reference images (top row) are translated to three before-makeup images (left column). Nine generated images are shown in the middle.

4. RELATED WORK

4.1 Style Transfer

Style transfer aims to combine content and style from different images. To achieve this goal, [8] proposed a method that generated a reconstruction image by minimizing the content and the style re-construction loss. To control more information like color, scale and spatial location, an improved approach was presented in [9], where perceptual factors were introduced. The methods mentioned above could produce high-quality results but require heavy computation. [13] proposed a feed-forward

network for style transfer with less computation and approximate quality.

4.2 Generative Adversarial Networks

Generative Adversarial Networks[10] (GANs) is one of the generative models, consisting a discriminator and a generator. GAN has been widely used in computer vision tasks due to its ability of generating visually realistic images. [17] presented a generative adversarial network for image super resolution. [6] employed conditional GAN[25] to solve particular eye in-painting problem. [27] trained adversarial models on synthetic images for improving the realism of them. [34] even enabled to incorporate user interactions to present real-time image editing, where GAN was leveraged to estimate the image manifold

4.3 GAN for Image-to-Image Translation

Most existing researches on image-to-image translation aim to learn a mapping from source domain to target domain. Recently, there are some promising works[4, 12, 35] applying GAN to this field. [12] proposed a so-called pix2pix framework, which could synthesize images from label maps and reconstruct objects from edge images. To solve the problem of lacking paired images for training, [22] proposed a model whose generators were bounded with weight-sharing constraints to learn a joint distribution. [35][14] presented cycle consistency loss to regularize the key attributes between inputs and translated images. StarGAN[4] even solved problem of mapping among multiple domains within one single generator. Specially, [15] introduced an encoder working with GAN for image attribute transfer.

5. BeautyGAN

The goal is to realize facial makeup transfer between a reference makeup image and a non-makeup image on instance-level. Consider two data collections, $A \subset \mathbb{R}^{H \times W \times 3}$ referring to non-makeup image domain and $B \subset \mathbb{R}^{H \times W \times 3}$ referring to makeup image domain with various makeup styles on. We simultaneously learn the mapping between two domains, denoted as $G: A \times B \rightarrow B \times A$, where '×' represents Cartesian product. That is to say, given two images as inputs: a source image $I_{src} \in A$ and a reference image $I_{ref} \in B$, the network is expected to generate an after-makeup image $I_{src}^B \in B$ and an anti-makeup image $I_{ref}^A \in A$, denoted as $(I_{src}^B, I_{ref}^A) = G(I_{src}, I_{ref})$. I_{src}^B synthesizes the makeup style of I_{ref} while preserving the face identity of I_{src} , and I_{src}^A refines makeup removal from I_{ref} . The fundamental problem is how to learn instance-level correspondence, which should ensure the makeup style consistency between result image I_{src}^B and reference images I_{ref} . Note that there is no paired data for training. To address the issue, we introduce pixel-level histogram loss acted on different cosmetics. In addition, perceptual loss has been employed to maintain face identity and structure. Then we can transfer the exact makeup to the source image without the change of face structure. The proposed method is based on Generative Adversarial

Networks[10], and it is convenient to integrate all loss terms into one full objective function. Adversarial losses help generate visually pleasant images and refine the correlation among different cosmetics. The details of loss functions and network architectures are shown below.

5.1 Methodology

Please refer to Figure 3. Consider two data collections, $A \subset \mathbb{R}^{H \times W \times 3}$ referring to non-makeup image domain and $B \subset \mathbb{R}^{H \times W \times 3}$ referring to makeup image domain. We simultaneously learn the mapping between two domains, denoted as $G: A \times B \rightarrow B \times A$, where '×' represents Cartesian product.

The overall framework consists of one generator G and two discriminators: D_A, D_B . In the formulation $(I_{src}^B, I_{ref}^A) = G(I_{src}, I_{ref})$, G accepts two images, $I_{src} \in A$ and $I_{ref} \in B$, as inputs and generates two translated images as outputs, $I_{src}^B \in B$ and $I_{ref}^A \in A$.

5.2 Objective Function

D_A and D_B contain only adversarial losses. D_A and D_B distinguish the generated image I_{ref}^A and I_{src}^B from 1 samples in set A and set B, respectively, given by:

$$\begin{aligned}\mathcal{L}_{D_A} &= \mathbb{E}_{I_{src}} [\log D_A(I_{src})] + \mathbb{E}_{I_{src}, I_{ref}} [\log (1 - D_A(I_{ref}^A))] \\ \mathcal{L}_{D_B} &= \mathbb{E}_{I_{ref}} [\log D_B(I_{ref})] + \mathbb{E}_{I_{src}, I_{ref}} [\log (1 - D_B(I_{src}^B))]\end{aligned}$$

The full objective function of generator G contains four types of losses: adversarial loss, cycle consistency loss, perceptual loss, and makeup constrain loss,

$$\mathcal{L}_G = \alpha \mathcal{L}_{adv} + \beta \mathcal{L}_{cyc} + \gamma \mathcal{L}_{per} + \mathcal{L}_{makeup}$$

where $\mathcal{L}_{adv} = \mathcal{L}_{D_A} + \mathcal{L}_{D_B}$

The perceptual loss:

The perceptual loss between input images I_{src} , I_{ref} and output images I_{src}^B , I_{ref}^A are expressed as:

$$\begin{aligned}\mathcal{L}_{per} &= \frac{1}{C_l \times H_l \times W_l} \sum_{ijk} E_l \\ E_l &= [F_l(I_{src}) - F_l(I_{src}^B)]_{ijk}^2\end{aligned}$$

where F_{ijk}^l is the activation of the i_{th} filter at position $<j, k>$ in l_{th} layer for an image x , $F_l(x)$, denotes the corresponding feature maps in l_{th} layer on VGG, where $F_l \in R^{C_l \times H_l \times W_l}$. C_l is the number of feature maps, H_l and W_l are height and width of each feature map, respectively.

The cycle consistency loss:

In order to maintain background information, we also introduce cycle consistency loss. When the output images are passed into the generator, it is supposed to generate images as close as the original input images. This procedure can be expressed as

$$(I_{src}, I_{ref}) \rightarrow G(I_{src}, I_{ref}) \rightarrow G(G(I_{src}, I_{ref})) \approx (I_{src}, I_{ref})$$

The loss function is formulated as

$$\mathcal{L}_{cyc} = \mathbb{E}_{I_{src}, I_{ref}} [dist(I_{src}^{rec}, I_{src}) + dist(I_{ref}^{rec}, I_{ref})]$$

where $(I_{src}^{rec}, I_{src}) = G(G(I_{src}, I_{ref}))$. The distance function $dist(\cdot)$ could be chosen as L_1 norm, L_2 norm, or other metrics.

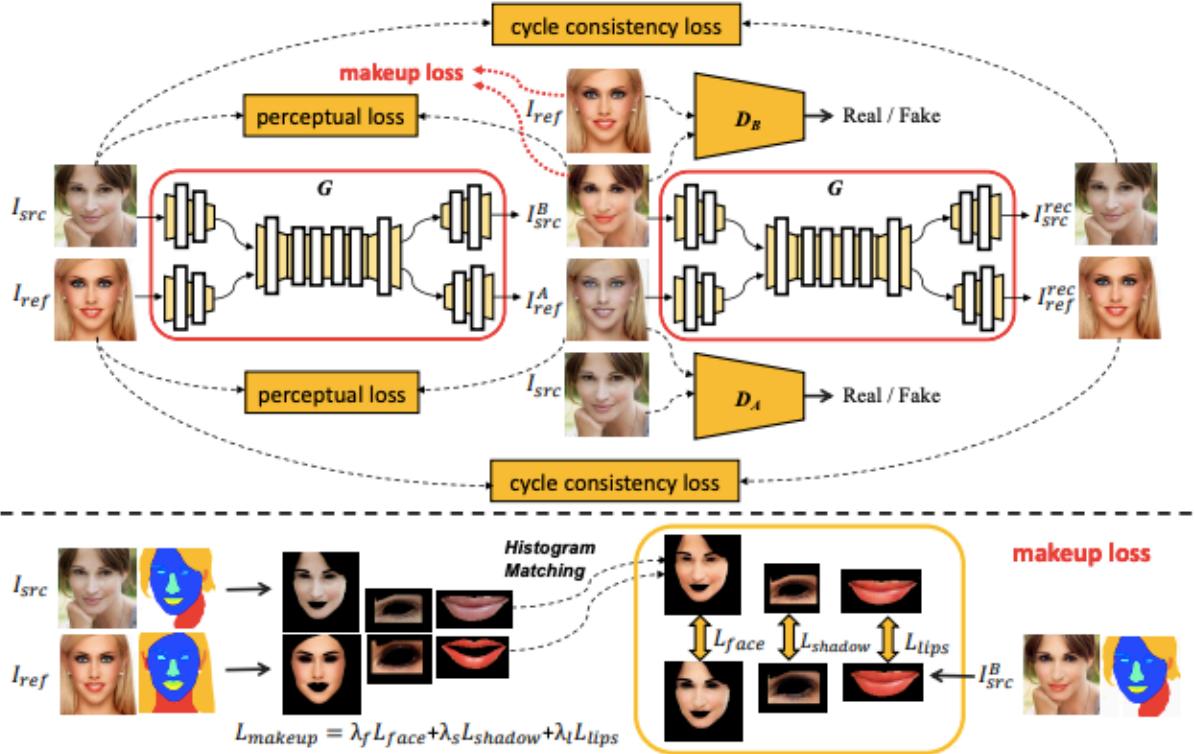


Figure 3: Framework of the proposed BeautyGAN [2].

The upper pipeline shows the overall system. G accepts two images as inputs: non-makeup image I_{src} , reference makeup image I_{ref} , and generates two outputs: transferred makeup image I_{src}^B , antimakeup image I_{ref}^A . The generated images are fed into the same G to build up reconstruction results: I_{src}^{rec} , I_{ref}^{rec} . There are four loss terms for training G : cycle consistency loss, perceptual loss, adversarial loss (denoted as D_A and D_B) and makeup loss. The lower pipeline shows the details of makeup loss. It consists of three local histogram loss terms acted on face, eye shadow and lips, respectively. We first utilize face parsing model to separate each cosmetic region of I_{src} , I_{ref} , I_{src}^B . Then, for each region, we employ histogram matching between I_{src} and I_{ref} to obtain a histogram remapping facial region as ground truth. The local loss term calculates pixel-level differences between such ground truth and corresponding cosmetic region of I_{src}^B .

images as close as the original input images. This procedure can be expressed as

$$(I_{src}, I_{ref}) \rightarrow G(I_{src}, I_{ref}) \rightarrow G(G(I_{src}, I_{ref})) \approx (I_{src}, I_{ref})$$

The loss function is formulated as

$$\mathcal{L}_{cyc} = \mathbb{E}_{I_{src}, I_{ref}} [\text{dist}(I_{src}^{rec}, I_{src}) + \text{dist}(I_{ref}^{rec}, I_{ref})]$$

where $(I_{src}^{rec}, I_{src}) = G(G(I_{src}, I_{ref}))$. The distance function $\text{dist}(\cdot)$ could be chosen as L_1 norm, L_2 norm, or other metrics.

Makeup loss:

Makeup loss is integrated by three local histogram losses acted on lips, eye shadows, and face regions, respectively:

$$\mathcal{L}_{makeup} = \lambda_l \mathcal{L}_{lips} + \lambda_s \mathcal{L}_{shadow} + \lambda_f \mathcal{L}_{face}$$

where λ_l , λ_s , λ_f are weight factors.

We multiply images with their corresponding binary masks and process spatially histogram matching between result image I_{src}^B and reference image I_{ref} . Formally, we define local histogram loss as

$$\mathcal{L}_{item} = \|I_{src}^B - HM(I_{src}^B \circ M_{item}^1, I_{ref}^2 \circ M_{item}^2)\|_2$$

$$M^1 = FP(I_{src}^B), M^2 = FP(I_{ref})$$

Here, \circ denotes element-wise multiplication, and item are in set of $\{\text{lips}, \text{shadow}, \text{face}\}$.

6. Instance-level Makeup Transfer

To further encourage the network to learn instance-level makeup transfer, it is essential to add constraints on makeup style consistency. We observe that facial makeup could be visually recognized as color distributions. No matter lipsticks, eye shadows or foundations, the makeup process could be mainly understood as color changing. There are various color transfer methods that can be found in the survey [7]. We employ Histogram Matching ($HM(x, y)$), a straight forward method, and introduce additional histogram loss on pixel-level, which restricts the output image and the reference image to be identical in makeup style.

Histogram loss:

If we directly adopt MSE loss on pixel-level histograms of two images, the gradient will be zero, owing to the indicator function, thus makes no

contribution to optimization process. Therefore, we adopt histogram matching strategy that generates a ground truth remapping image in advance. Suppose that we want to calculate histogram loss on pixels between original image x and reference image y , we should first perform histogram matching on x and y to obtain a remapping image $HM(x, y)$, which has the same color distribution as y but still preserves content information as x . After we get $HM(x, y)$ it is convenient to utilize the MSE loss between $HM(x, y)$ and x , then back-propagate the gradients for optimization.

Face Parsing.

Instead of utilizing histogram loss over the entire image, we split the makeup style into three important components – lipsticks, eye shadow, and foundation. We employ the face parsing learned model of “**shape_predictor_68_face_landmarks.dat.bz2**” to obtain face guidance mask as $M = FP(x)$. For each input image x , pre-trained face parsing model would generate an index mask M denoting several facial locations, including lips, eyes, face skin (corresponds to foundation), hair, background, and so on. At last, for each M , we track different labels to produce three corresponding binary masks, representing for cosmetics spatiality: M_{lips} , M_{eyes} , M_{face} . It is important to note that eye shadows are not annotated on M , because the before-makeup images have no eye shadows. But we expect the result image I_{src}^B , to have similar eye shadow color and shape as reference image I_{ref} . According to eyes mask M_{eyes} , we calculate two rectangle areas enclosing eye shadows and then exclude eyes regions, some hair and eyebrow regions in between. Thus we could create a specific binary mask representing for eye-shadows M_{shadow} , show right below on Figure 4.

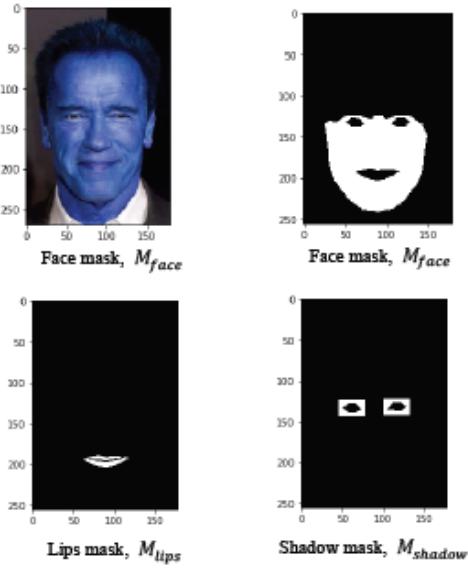


Figure 4:
Extracted masks by face parsing learned model

7. Dataset

We collect a new facial makeup dataset consisting of 3834 female images in total, with 1,115 non-makeup images and 2,719 makeup images. We refer to this dataset as the Makeup Transfer (**MT**) dataset. It includes some variations in race, pose, expression, and background clutter. Plenty of makeup styles have been assembled, including smoky-eyes makeup style, flashy makeup style, Retro makeup style, Korean makeup style, and Japanese makeup style, varying from subtle to heavy. Specifically, there are some nude makeup images, as for convenience, have been classified into non-makeup category.



Figure 5: The images in Dataset. The upper row images are the non-make up images, the bottom row images are make-up images.

8. EXPERIMENTS

In this section, we depict the network architecture and training setting. All the experiments apply the same **MT** dataset we release. We compare performances of our method and some other baselines from both qualitative and quantitative perspectives. And we further give a thorough analysis on components of BeautyGAN.

The `relu_4_1` feature layer of **VGG16**[28] network is applied for identity preserving. Such VGG16 is pre-trained on ImageNet and has parameters fixed all through training process. The parameters in equations 3 and 9 are: $\alpha = 1$, $\beta = 10$, $\gamma = 0.005$, $\lambda_l = 1$, $\lambda_s = 1$, $\lambda_f = 0.1$. We train the network from scratch using Adam[16] with learning rate of 0.0002 and batch size of 1.

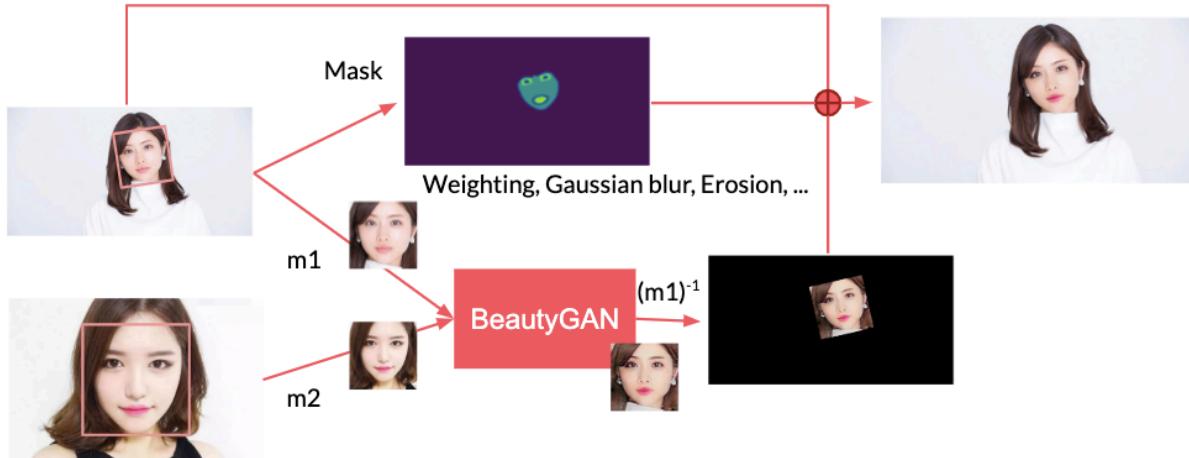


Figure 5: Achitecture of proposed method



Figure 5: The images in Dataset. The upper row images are the non-make up images, the bottom row images are make-up images.

9. RESULT OF BEAUTYGAN

The zoom in performance of eye makeup and lipsticks transfer are shown below on the right. We also listed one set of our generated results on the right. The last column shows the makeup transfer results.

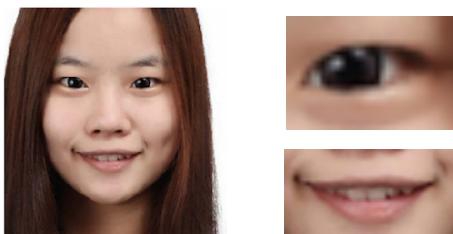


Figure 6: Non-makeup image.



Figure 7: Reference (makeup) image



Figure 8: Fake image by BeautyGAN

10. ACHITECTURE OF OUR METHOD

Our proposed method has 4 steps. Each step are explained as follow. We upload the source code in github(https://github.com/sakurayamaki/Generating-makeup-image-with-BeautyGAN-using-web-camera/tree/master/source_code)Please try it and enjoy by them.

STEP1: Capture image or videos from web camera am on PC and save them as jpeg or mp4 format files.



Figure 9: Non-makeup image captured by web camera

STEP2: Select one of our favor make up styled images. We prepared 5 make-up styles.

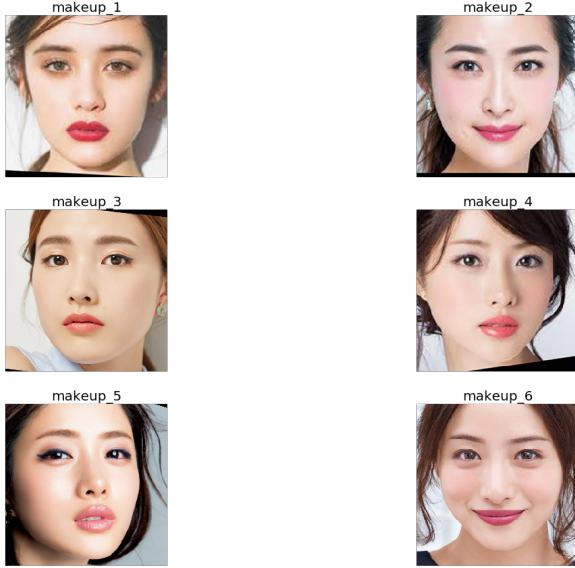


Figure 10: 5 kind of make-up styles

STEP3: Generate a Fake image from two images with BeautyGAN using a trained model. The steam of generating fake image is shown in Fig. 5. In order to recognize face clearly and generate detailed fake image, we extract the part of face before training with BeautyGAN. And then train image with model using extracted non-makeup image, finally we get a result image which generated face part are embedded.

STEP4: Save the Fake image and plot it on the screen.



Figure 11: Selected makeup image



Figure 12: Generated image on BeautyGAN[2] using non-makeup image captured by web camera

11. Discussion

Discuss the images generated by BeautyGAN. First of all, Face Parsing was able to detect the face part. However, as a bad point, there is a part which can not be detected in the mouth part or the outline of the eye part is

blurred. Next we will discuss the makeup feature. As a result, the skin color of the face of the makeup image is acquired and applied to the image acquired by the web camera. However, on the other hand, the color of the eye shadow has not been applied to the image acquired by the webcam. Also, with regard to the color of the lips, the color of the lips of the webcam appears lighter than the color of the lipstick in the makeup image. There is still room for improvement.

However, the result of applying BeautyGAN to the image collected from internet reflects makeup style very well. Cosmetic style transfer tends to be improved in bright environments with light.



Figure 13: The result of BeautyGAN on the image collected from internet.

12. Future work

There are 3 remained work for now.

- (1) The result of makeup image using web camera isn't good since the skin color is dark which is not fit with the bight makeup part. So intensity transfer such as gamma correction is needed as pre-processing.
- (2) The whole pipeline takes about 0.5 sec for a single frame, not fast enough for real-time usage.
- (3) Replace Dlib face parsing with NN based method to improve the result. Specifically, according to our method, as shown in Fig. 4, the result of segmented the face part does not include the forehead of the face skin part.
- (4) Histogram matching only preserve the make-up color tone, but the detail is ignored.

12. Reference

- [1] T. T. Li, R. H. Qian, C. Dong, S. Liu, Q. Yan, W. W. Zhu, and L. Lin, "BeautyGAN: Instance-Level Facial Makeup Transfer with Deep Generative Adversarial Network,"

- http://iisi-group.com/pdf/BeautyGAN-camera-ready_2.pdf, 2018.
- [2] Cunjian Chen, Antitza Dantcheva, and Arun Ross. 2013. Automatic facialmakeup detection with application in face recognition. In Biometrics (ICB), 2013 International Conference on. IEEE, 1–8.
 - [3] Cunjian Chen, Antitza Dantcheva, Thomas Swearingen, and Arun Ross. 2017. Spoofing faces using makeup: An investigative study. In Identity, Security and Behavior Analysis (ISBA), 2017 IEEE International Conference on. IEEE, 1–8.
 - [4] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2017. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. arXiv preprint arXiv:1711.09020 (2017).
 - [5] Antitza Dantcheva, Cunjian Chen, and Arun Ross. 2012. Can facial cosmetics affect the matching accuracy of face recognition systems?. In Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on. IEEE, 391–398.
 - [6] Brian Dolhansky and Cristian Canton Ferrer .2017. EyeIn-PaintingwithExemplar Generative Adversarial Networks. arXiv preprint arXiv:1712.03999 (2017).
 - [7] Hasan Sheikh Faridul, Tania Pouli, Christel Chamaret, Jürgen Stauder, Alain Tréneau, Erik Reinhard, et al. 2014. A Survey of Color Mapping and its Applications. Eurographics (State of the Art Reports) 3 (2014)
 - [8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576 (2015).
 - [9] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. 2017. Controlling perceptual factors in neural style transfer. In IEEE Conference on Computer Vision and Pattern Recognition.
 - [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Advances in neural information processing systems. 2672–2680.
 - [11] Dong Guo and Terence Sim. 2009. Digital face makeup by example. In Computer Vision and Pattern Recognition, 2009. IEEE Conference on. IEEE, 73–79.
 - [12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2016. Image- to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004 (2016).
 - [13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real- time style transfer and super-resolution. In European Conference on Computer Vision. Springer, 694–711.
 - [14] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relation swith generative adversarial networks. arXiv preprint arXiv:1703.05192 (2017).
 - [15] Taeksoo Kim, Byoungjip Kim, Moonsu Cha, and Jiwon Kim. 2017. Unsupervised visual attribute transfer with reconfigurable generative adversarial networks. arXiv preprint arXiv:1707.09798 (2017).
 - [16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
 - [17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2016. Photo-realistic single image super-resolution using generative adversarial network. arXiv preprint arXiv:1609.04802 (2016).
 - [18] Chuan Li and Michael Wand. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In European Conference on Computer Vision. Springer, 702–716.
 - [19] Chen Li, Kun Zhou, and Stephen Lin. 2015. Simulating makeup through physics- based manipulation of intrinsic image layers. In IEEE Conference on Computer Vision and Pattern Recognition. 4621–4629.
 - [20] Yi Li, Lingxiao Song, Xiang Wu, Ran He, and Tieniu Tan. 2017. Anti-Makeup: Learning A Bi-Level Adversarial Network for Makeup-Invariant Face Verification. arXiv preprint arXiv:1709.03654 (2017).
 - [21] Jing Liao, Yuan Yao, Lu Yuan, Gang Hua, and Sing Bing Kang. 2017. Visual attribute transfer through deep image analogy. ACM Transactions on Graphics (TOG) 36, 4 (2017), 120.
 - [22] Ming-Yu Liu and Oncel Tuzel. 2016. Coupled generative adversarial networks. In advances in neural information processing systems. 469–477.
 - [23] Si Liu, Xinyu Ou, Ruihe Qian, Wei Wang, and Xiaochun Cao. 2016. Makeup like a superstar: deep localized makeup transfer network. In the Association for the Advance of Artificial Intelligence. AAAI Press, 2568–2575.
 - [24] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. 2016. Multi-class Generative Adversarial Networks with the L2 Loss Function. CoRR abs/1611.04076 (2016). arXiv:1611.04076 <http://arxiv.org/abs/1611.04076>
 - [25] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014).
 - [26] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018).
 - [27] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. 2016. Learning from simulated and unsupervised images through adversarial training. arXiv preprint arXiv:1612.07828 (2016).
 - [28] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR abs/1409.1556 (2014).
 - [29] Wai-Shun Tong, Chi-Keung Tang, Michael S Brown, and Ying-Qing Xu. 2007. Example-based cosmetic

- transfer. In Computer Graphics and Applications, 2007. PG'07. 15th Pacific Conference on. IEEE, 211–218.
- [30] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. CoRR abs/1607.08022 (2016). arXiv:1607.08022 <http://arxiv.org/abs/1607.08022>
- [31] Shuyang Wang and Yun Fu. 2016. Face Behind Makeup. In the Association for the advance of Artificial Intelligence. 58–64.
- [32] Zhen Wei, Yao Sun, Jinqiao Wang, Hanjiang Lai, and Si Liu. 2017. Learning Adaptive Receptive Fields for Deep Image Parsing Network. In IEEE Conference on Computer Vision and Pattern Recognition. 2434–2442.
- [33] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In IEEE Conference on Computer Vision and Pattern Recognition. 2881–2890.
- [34] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. Generative visual manipulation on the natural image manifold. In European Conference on Computer Vision. Springer, 597–613.
- [35] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593 (2017).

DIGITAL IMAG FINAL PROJI

¹ *Sakura Yamaki (八卷櫻),¹*

¹ Department of Computer Scie
National Taiwa
E-mail: r07922153@ntu.edu.t

ABSTRACT

Facial makeup transfer aims to translate the makeup style from a given reference makeup face image to another non-makeup one while preserving face identity. Such an instance-level transfer problem is more challenging than conventional domain-level transfer tasks, especially when paired data is unavailable. Makeup style is also different from global styles (e.g., paintings) in that it consists of several local styles/cosmetics, including eye shadow, lipstick, foundation, and so on. Extracting and transferring such local and delicate makeup information is infeasible for existing style transfer methods. So we use the issue by incorporating both global domain-level loss and local instance-level loss in an dual input/output Generative Adversarial Network, called BeautyGAN[2]. BeautyGAN generates and outputs an image close to the makeup style of the image which show make-up style on the video image which is capture from web camera. We upload the source code in github (https://github.com/sakurayamaki/Generating-makeup-image-with-BeautyGAN-using-web-camra/tree/master/source_code). Please try it and enjoy by them.

Keywords: Facial makeup transfer, *GAN*, *Live streaming*.

1. INTRODUCTION

Live broadcasting is one of the most popular industries recently. People use live video to share their lives and some even make money from the fast growing market and also many paper concern to make-up transfer system such as [1, 2, 3, 35] are published recently. During broadcasting, face filters are often used to make you look more stylish and appealing in the live video. However, existing services only provide few pre-defined styles and might not be suitable for anyone. In this project, the ultimate goal is to make a flexible face filtering system that allows users to customize their styles by providing images or their favorite styles, and our system is able to extract the styles, such as, make-up, skin tone, and so on, and apply them to the user faces.

2. PROBLEM DEFINITION

Input a target and example face, we would like to develop a model that extracts the style from the example and output the target face with the example style applied.