Deep Face Rotation in the Wild

Shohei Morikawa
Tokyo Institute of Technology
Tokyo, Japan
shohei@img.cs.titech.ac.jp

Suguru Saito
Tokyo Institute of Technology
Tokyo, Japan
suguru@c.titech.ac.jp

Figure 1: Turnaround face images generated from the leftmost images via a latent code space with a pose condition parameter. The source images in the leftmost are generated by our model with random latent codes.

ABSTRACT

Generating face images in various directions from an image will be useful to create avatars in VR. In this paper, we introduce a new deep generative model to generate turnaround face images from an image via a latent code space with a parameter. The model was learned with a large scale image dataset annotated with attributes but not including exact target images.

CCS CONCEPTS

• **Computing methodologies** → *Image processing*; *Neural networks*; *Object recognition*;

KEYWORDS

Image generation, Neural networks, Face rotation image

ACM Reference Format:
Shohei Morikawa and Suguru Saito. 2018. Deep Face Rotation in the Wild. In *VRST 2018: 24th ACM Symposium on Virtual Reality Software and Technology (VRST '18), November 28-December 1, 2018, Tokyo, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3281505.3281606

1 INTRODUCTION

Changing face direction increases the expression for avatars in VR. In this paper, a new neural network(NN) technique is applied to the problem of changing the face direction of an input image with continuous parameters.

Comparing with the previous work, which is an transformation for a face direction to discrete specific face directions, or which requires special dataset that contains several images for a person to be shot from various view directions in controlled conditions, our NN is trained with a dataset, which consists of face images shot in uncontrolled condition but labeled with semantic attributes, and allows to transform the face dirction of an image according to continuous parameters.

2 ARCHITECTURE

The main of our NN is composed of a Conditional VAE and a GAN as shown in Figure 2. The sub-networks E, G, P and D_2 consist of one convolution layer with stride-2, three residual blocks with stride-2 and two fully connected layers. For the sub-network D_1, refer to Adversarial Variational Bayes[2]. The VGG is a pre-trained VGG16 network on the ImageNet. In P, D_1 and D_2, a global average pooling layer is adopted instead of the first fully connected layer. Leaky ReLU is used as the activation function in the sub-networks except for P, whereas ReLU is used for P.

2.1 Losses

For the image reconstruction loss function L_{rec}, the L_2 difference between feature output from the the first 3 blocks in VGG16 are adopeted as,

$$L_{rec} = \sum_n ||\Phi_n(x) - \Phi_n(G(z_x, y_x^p))||_2, \quad (1)$$

where $\Phi_n(x)$ is the output of the n-th middle layer in the VGG16 network for an image x, G generates an image for a latent code and a pose condition, and z_x and y_x^p are the latent code and the the pose condition, which are corresponding to the image x, respectively.

The loss functions L_{D_1} and L_{D_2} are defined for the two discriminators, which are learned for the latent code distribution and the training data distribution respectively. The loss functions

VRST '18, November 28-December 1, 2018, Tokyo, Japan

ACM ISBN 978-1-4503-6086-9/18/11.
https://doi.org/10.1145/3281505.3281606

$L_{codeGAN}$ and L_{imgGAN} are used to train the sub-network E and G to deceive D_1 and D_2.

$$L_{imgGAN} = -\mathbb{E}_{z\sim p_z(z)}[logD_2(G(z, y^p))] \quad (2)$$

$$L_{codeGAN} = -\mathbb{E}_{x\sim p_{data}(x)}[logD_1(z_x)] \quad (3)$$

$$L_{D_2} = -\mathbb{E}_{x\sim p_{data}(x)}[logD_2(x)] - \mathbb{E}_{z\sim p_z(z)}[log(1 - D_2(G(z, y^p)))] \quad (4)$$

$$L_{D_1} = -\mathbb{E}_{z\sim p_z(z)}[logD_1(z)] - \mathbb{E}_{x\sim p_{data}(x)}[log(1 - D_1(z_x))] \quad (5)$$

where the prior distribution of latent space p_z is set to $\mathcal{N}(0, 1)$.

The predictor is the network estimating the pose and attributes of an input image. Those output are used by the following two loss functions. In order to train this network, the L_{pred} is defined by Equation6 which consists of the L_2 difference between inferenced and labeled poses and the sigmoid cross entropy for the multi label classification of the attributes.

$$L_{pred} = ||y_x^p - P_{pose}(x)||_2 - y_x^a logP_{attr}(x) - (1 - y_x^a)log(1 - P_{attr}(x)) \quad (6)$$

where y_x^a is the attributes that an image x has.

The condition loss function L_{pose} lets the estimated pose for a generated image with a latent code z and a condition y^p be the same.

$$L_{pose} = ||y^p - P_{pose}(G(z, y^p))||_2 \quad (7)$$

Images generated from a latent code should be for the same person for any condition value. As a lesser strict evaluation than a complete identity judgment, the following attribution loss function L_{attr} is introduced with an assumption that a person has the same attributes while changing the face direction.

$$L_{attr} = -P_{attr}(G(z, y_1^p))logP_{attr}(G(z, y_2^p)) - (1 - P_{attr}(G(z, y_1^p)))log(1 - P_{attr}(G(z, y_2^p))) \quad (8)$$

3 EXPERIMENTAL RESULTS

3.1 Dataset

For our experiment, we adopted CelebA [1], which is a large-scale dataset of more than 200K face color images annotated with 40 binary attributes. 160K images were selected randomly for the training and the rest were prepared for the test. Since there is no information for face orientation angles (yaw, pitch, roll) in the attributes of the dataset, those are additionally annotated with Hopenet [3] as a pose label, where 0 is the just frontal orientation.

3.2 Detail of Training

The images are resized as 64×64 pixel size. The value of color channels is normalized to $[-1, 1]$. The pose label is also normalized as $[-1, 1]$ and the 40 binary attributes are labeled as a 40 dimensional vector whose element is 1 or 0.

The mini batch size is set to be 64. All weights are initialized from a zero-centered normal distribution with a standard deviation of 0.02. We train the networks using Adam with $\alpha = 10^{-4}$, $\beta_1 = 0.5$, $\beta_2 = 0.999$.

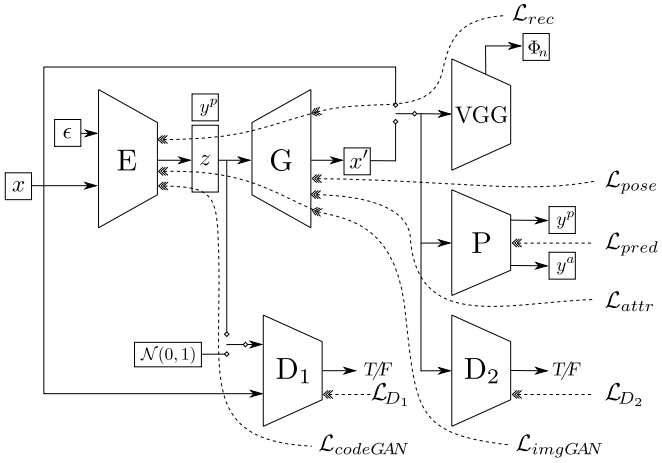

Figure 2: Architecture of our model

3.3 Generation

Figure 1 shows input and generated images. The input images in the leftmost were prepared by the G with sampled latent codes z from the prior distribution. The turnaround faces in each line were generated with the latent code z corresponding with the leftmost image x and the pose conditions y^p whose yaw value is -1 to 1 from left to right.

The generated face image quality and identity stability decrease as for y^p departing from 0, but when the absolute of y^p is smaller than a certain value, the identity is preserved. Please watch the appendix movie.

4 CONCLUSION

We propsed a framework to model a generator for synthesizing or transforming face images, whose face direction is controllable by a continuous parameter, using a dataset of face images shot in uncontralled conditions. The attribute loss function, which forces to preserve attributes of an image whatever conditional parameter changes, and the conditional parameter loss function, which forces for an output image to follow the condition allow to model the image generator without requiring a target image dataset.

REFERENCES

[1] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[2] Lars M. Mescheder, Sebastian Nowozin, and Andreas Geiger. 2017. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. *CoRR* abs/1701.04722 (2017). arXiv:1701.04722 http://arxiv.org/abs/1701.04722

[3] Nataniel Ruiz, Eunji Chong, and James M. Rehg. 2017. Fine-Grained Head Pose Estimation Without Keypoints. *CoRR* abs/1710.00925 (2017). arXiv:1710.00925 http://arxiv.org/abs/1710.00925