Facial Image Generation by Generative Adversarial Networks Using Weighted Conditions

Hiroki Adachi, Hiroshi Fukui, Takayoshi Yamashita, Hironobu Fujiyoshi
Chubu University, Kasugai, Aichi, Japan
{*ha618, fhiro*}*@mprg.cs.chubu.ac.jp,* {*yamashita, hf*}*@isc.chubu.ac.jp*

Keywords: Conditional Generative Adversarial Networks, CelebA Dataset

Abstract: CGANs are generative models that depend on Deep Learning and can generate images that meet given conditions. However, if a network has a deep architecture, conditions do not provide enough information, so unnatural images are generated. In this paper, we propose a facial image generation method by introducing weighted conditions to CGANs. Weighted condition vectors are input in each layer of a generator, and then a discriminator is extend to multi-tasks so as to recognize input conditions. This approach can step-by-step reflect conditions inputted to the generator at every layer, fulfill the input conditions, and generate high quality images. We demonstrate the effectiveness of our method in both subjective and objective evaluation experiments.

1 INTRODUCTION

Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have received great research interests recently because this method is able to generate images or sentences using random noise vectors. Therefore, many methods have been proposed on the basis of learning techniques of GANs. Conditional GANs (CGANs) (Mirza and Osindero, 2014)(Reed et al., 2016) can generate images that fulfil certain conditions by inputting class labels, text, and so on as conditions. GANs and CGANs are generally constructed by Multi Layer Perceptrons (MLPs), which causes these have various problems such as unstable training, making it difficult to generate high-quality images. High-quality images can be generated by Deep Convolutional GANs (DCGANs) (Radford et al., 2016) and by replacing fully connected layers of CGANs with convolution layers (Conditional DCGANs) (Gauthier, 2014). Specifically, DCGANs are able to make training more stable by adding various training techniques. Recently proposed methods include unsupervised learning that can generate images like CGANs as an auxiliary task (Chen et al., 2016), and a method to improve the quality of generated images (Augustus et al., 2017). Moreover, the latest state-of-the-art method, Progressive Growing GANs (PGGANs) (Karras et al., 2018), can generate high quality, natural-looking images by using a hierarchical training process.

However, CGANs and Conditional DCGANs have a problem in that inputted conditions vanish near the output layer, so the generated images become unnatural when deep architecture networks such as PGGANs are used because conditions are inputted in only the first layer. Therefore, in this paper, we propose a facial image generation method by introducing weighted conditions to CGANs. The proposed method generates images that stepwisely reflect conditions by inputting weighted conditions to a generator. Additionally, to reflect conditions further after adversarial learning of the generator, a discriminator expands multi-tasks so as to recognize conditions input to the generator. Furthermore, we construct an encoder to extract the feature quantity of the input image and propose a learning method that can reconstruct images by inputting the extracted feature quantity to the generator of the proposed method.

2 RELATED WORKS

2.1 Generative Adversarial Networks

GANs (Goodfellow et al., 2014) are generative models using Deep Learning that consist of two networks: a generator and a discriminator. The generator generates an image that deceives the discriminator by using noise vectors as input. The discriminator accurately

classifies between inputted real images and generated images. The objective function of GANs is given as

$$\min_G \max_D V(D,C) = \mathbb{E}_{\boldsymbol{x}\sim P_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim P(\boldsymbol{z})}[\log(1-D(\tilde{\boldsymbol{x}}))], \quad (1)$$

where $\boldsymbol{z} \in \mathbb{R}^{100}$ is noise vectors sampled from distribution $p(\boldsymbol{z})$ such as $\mathcal{N}(0,I)$ or $\mathcal{U}[-1,1]$, $\tilde{\boldsymbol{x}}$ is images generated by the generator, and $\boldsymbol{x}$ is real images. By adversarial learning of the generator and discriminator, images not included in training samples can be generated. Also, unlike a Variational Autoencoder (VAE) (Kingma and Welling, 2014), GANs are able to generate images that are not blurry because they do not calculate error in pixel units. Vanilla GANs have difficulty generating specific images because only noise vectors are inputted. CGANs (Mirza and Osindero, 2014) are able to generate images that fulfill conditions by using conditions such as class labels and text corresponding to images. The objective function of CGANs is given as

$$\min_G \max_D V(D,C) = \mathbb{E}_{\boldsymbol{x}\sim P_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim P(\boldsymbol{z})}[\log(1-D(\hat{\boldsymbol{x}}|\boldsymbol{y}))], \quad (2)$$

where $\hat{\boldsymbol{x}}$ is $G(\boldsymbol{z}|\boldsymbol{y})$ obtained by inputting the noise vector $\boldsymbol{z}$ and conditions $\boldsymbol{y}$ to the generator. Vanilla GANs or CGANs have difficulty generating clear images because of the way the MLP is configured. To overcome this problem, DCGANs (Radford et al., 2016), Conditional DCGANs (Gauthier, 2014), and PGGANs (Karras et al., 2018) in which Deep Convolutional Neural Network (DCNN) (Yann et al., 1998) that have a convolutional layer and Batch Normalization (Sergey and Christian, 2015) are introduced have been proposed and are able to generate high-quality images. In particular, PGGANs can stably generate high-resolution natural images by first generating global information, gradually adding a convolutional layer to the network, generating detailed information, and imposing a penalty for errors called Wasserstein GANs Gradient Penalty (WGANs-GP) (Gulrajani et al., 2017).

2.2 Reconstruction Input Images

Many methods have been proposed that reconstruct input data with an encoder such as VAE. A Conditional Adversarial Autoencoder (CAAE) (Zhang et al., 2017) extracts rich feature vectors after inputting high-dimensional facial images to an encoder. Then, by inputting the condition of age in addition to the extracted feature vector to the generator, CAAE can change the input facial image to various ages. Bidirectional GANs (BiGANs) (Donahue et al., 2017)

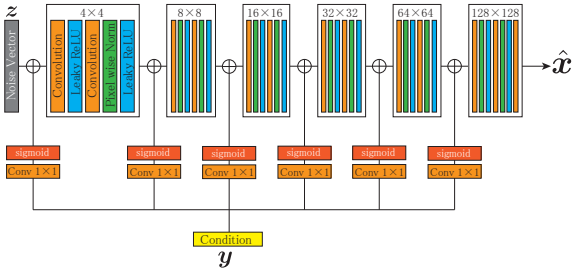

Figure 1: Generator adopted in proposed method

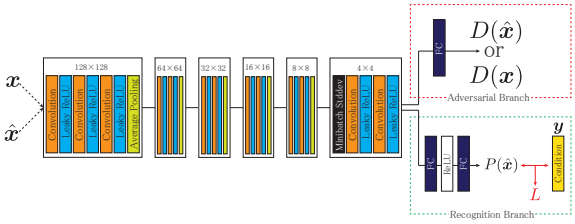

Figure 2: Discriminator adopted in proposed method

use a learning method that inputs not only generated images and real images but also the noise vector inputted to the generator in addition to features obtained from the encoder. α-GANs (Rosca et al., 2017) are similar to BiGANs but are separating networks that recognize the noise vector and the output of the encoder. Also, α-GANs add the L1 norm between the training data and generated data as reconstruction loss. These methods are able to generate clear images by adversarial learning.

3 PROPOSED METHOD: WEIGHTED CONDITIONS AND MULTI-TASK LEARNING

In this paper, we propose a facial image generation method that inputs weighted conditions to the generator and recognizes conditions in the discriminator. We also propose a method that reconstructs inputted images by using the encoder and the generator in the proposed method. First, we describe the learning manner of the generator in 3.1 and leaning manner of the multi-task discriminator in 3.2. Then we present the learning algorithm using the encoder and generator in 3.3.

3.1 Introduced Weight: Learning of Generator

In previous CGANs, conditions vanish near the output layer because the conditional vector $\boldsymbol{y} \in \{0,1\}$ is only in the input layer. Thus, the generator in the

proposed method inputs conditions to a hidden layer other than the input layer in like a skip connection. This approach can certainly reflect conditions until the near the output layer. In addition, previous facial image generation methods directly input the binary condition vector to the generator. On the other hand, the proposed method applies 1×1 convolution process and sigmoid function to the condition vector $\boldsymbol{y}$ expressed in binary and inputs its output to the generator. Therefore, we represent a continuous value $\boldsymbol{y}$ from 0 to 1 as a condition vector. Moreover, each condition can be weighted because the filter size of the convolutional process is 1×1. By weighting conditions, the proposed method is able to stepwisely reflect conditions in such a way as to whole the generator because the most suitable conditions can be reflected at the time of generation in each layer. Furthermore, we use Pixelwise Normalization instead of Batch Normalization. Pixelwise Normalization is a normalization method used in PGGANs that is able to improve the quality of generated images. Pixelwise Normalization is represented as

$$b_{x,y} = \frac{a_{x,y}}{\frac{1}{N}\sum_{j=0}^{N-1}(a_{x,y}^{j})^2 + \varepsilon}, \tag{3}$$

where N is the number of feature maps, $a_{x,y}$ and $b_{x,y}$ is the feature vector before and after and $\varepsilon = 10^{-8}$. This series of processes is indicated in Figure 1.

3.2 Multi-Task Discriminator

The discriminator inputs real or generated images and simultaneously considers inputted conditions to distinguish between the real images or generated ones are inputted to the discriminator, which simultaneously considers inputted conditions to distinguish between the images. The discriminator in our proposed method improves multi-tasks so as to recognize given conditions when the generator generates images. Figure 2 shows a multi-task network. The adversarial branch and recognition branch in Figure 2 represent a previous task of GANs and condition recognition, respectively. In CGANs and Conditional DCGANs, conditions are also given to the discriminator, but in the proposed method add the recognition branch. It is able to be considered alternative input conditions by minimizing the condition recognition error, which is computed by using the conditions inputted to the generator. Minibatch Stddev is the standard deviation for Mini Batch calibration. This proposed method at PGGANs is able to generate diverse images.

Condition recognition error is added to the objective function of previous CGANs. Thereby, adversarial learning of the generator reflects more conditions. The objective function of our proposed method is indicated as

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim P_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim P(\boldsymbol{z})}[\log(1-D(\tilde{\boldsymbol{x}})] \wedge \min L, \tag{4}$$

where L is condition recognition error. If a dataset of real facial images is used, our proposed method finds it difficult or impossible to recognize the images by using the softmax function and cross entropy error because multiple facial attributes in this dataset are represented in binary. When mean square error is used, the recognition branch of the number of attributes to be recognized is required and calculation cost is high. Hence, we calculate error by sigmoid cross entropy because we calculate recognition error of multiple facial attributes with a one the recognition branch.

3.3 Obtain Feature Vector: Encoder and Fine-Tunned Generator

Generative methods such as α-GANs and BiGANs use adversarial learning and an encoder and generate images without fine-tunned the generator. Therefore, generative methods frequently generate unclear images in initial learning. In addition, previous techniques are high cost because they require multiple networks to be updated. Thus, we propose a way of learning that uses an encoder and a fine-tuned generator. Clear facial images can be generated from initial learning using the fine-tuned generator, and our method generates images that maintain the identity of inputted images by inputting features obtained from the encoder to generator. Algorithm 1 details the proposed learning process, and Figure 3 is illustration of prior.

Both $\boldsymbol{f}$ and $\hat{\boldsymbol{f}}$ are features output from the Encoder, but the former is real images, and latter is generated images. Moreover, all $\mathcal{L}$ in Algorithm 1 are Mean Squared Error, but these errors are different. $\mathcal{L}_{real}$ is the error of real images and their reconstructions, $\mathcal{L}_{noise}$ is error of the noise vector and embedded features of image generated from the noise vector, and $\mathcal{L}_{fake}$ is error of reconstructed images and image generated from the noise vector. In our proposed learning algorithm fixes parameters of the generator and updates only the encoder.

4 EXPERIMENT

We evaluate the quality of facial images generated in the proposed method. Moreover, we evaluate the effectiveness of the multi-task Discriminator and weighted condition Generator.

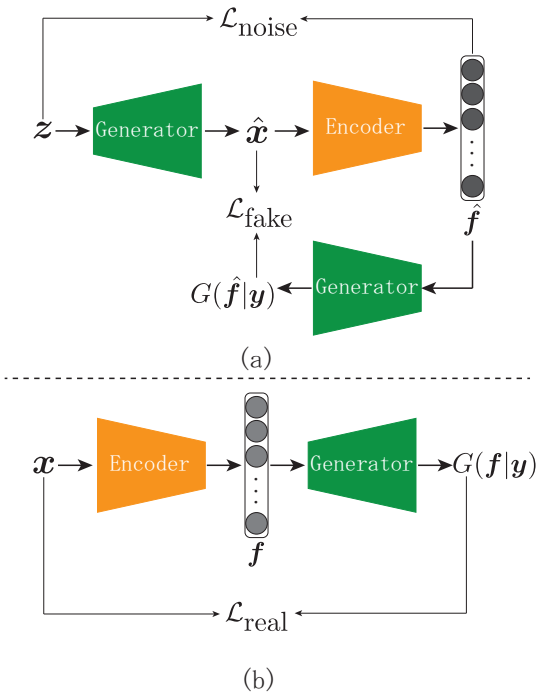

Figure 3: Training process using encoder and fine-tuned generator. (a) Reconstruction process using generated images from noise vector. (b) Reconstruction process using real images. Light gray and dark gray circles are feature vector which embedded real images and fake ones from, respectively.

Algorithm 1 Training process using encoder and fine-tuned generator. m is batch size and $\lambda = 0.1$.

for Number of training iterations **do**

• Sampling minibatch of m noise data, training data, and conditions $\boldsymbol{z}_m \in P(\boldsymbol{z})$, $\boldsymbol{x}_m \in P(\boldsymbol{x})$ and $\boldsymbol{y}_m \in P(\boldsymbol{y})$.

if Reconstruction of generated images from noise vector $\boldsymbol{z}$ **then**

$$\mathcal{L}_{noise} = \frac{1}{m}\sum_{i=1}^{m}[\boldsymbol{z} - E(\hat{\boldsymbol{x}}|\boldsymbol{y})]_i^2$$

$$\mathcal{L}_{fake} = \frac{1}{m}\sum_{i=1}^{m}\left[G(\boldsymbol{z}|\boldsymbol{y}) - G(\hat{\boldsymbol{f}}|\boldsymbol{y})\right]_i^2$$

else if Reconstruction using real images **then**

$$CH = \{R, G, B\}$$

$$\mathcal{L}_{real} = \sum_{i \in CH}\left[\frac{1}{m}\sum_{j=0}^{m}(\boldsymbol{x} - G(\boldsymbol{f}|\boldsymbol{y}))_j^2\right]_i$$

end if

$$\mathcal{L} = exp(\lambda(\mathcal{L}_{noise} + \mathcal{L}_{fake} + \mathcal{L}_{real}))$$

• Updating the encoder by using Adam optimizer.

end for

4.1 Experimental details

In this experiment, facial images generated using the conventional methods (Conditional DCGANs and Conditional PGGANs) and DCGANs and PGGANs using the proposed method (Weighted Condition DCGANs and Weighted Condition PGGANs) are compared. We use CelebA Dataset (Ziwei et al., 2015) which contains at 200,000 facial images during training of every methods. For the condition, a five-dimensional condition vector $\boldsymbol{y}$ is created using five attributes (Male, Bangs, Eyeglasses, Goatee, and Smiling) of 40 kinds of face attributes given to each image of CelebA Dataset. Moreover, the noise vector of 512-dimension sampling from a normal distribution is input to the generator. We compare the quality of generated images in objective and subjective evaluations. In the objective evaluation, Inception Score and Fréchet Inception Distance (FID) are used. The Inception Score is the average result of 10 evaluations. In the subjective evaluation, we use 150 images every method and in 21 subjects evaluate generated images in terms the quality and condition fulfilment. We create the simple user interface for the subjective evaluation.

4.2 Experimental Results

Figure 4 shows facial images generated by each method. In the visual evaluation, images generated by all method are able to clearly show faces, and whether the generated facial images reflected inputted conditions is determined. Figure 4 (a) to (d) show all methods were able to generate images of the same quality. Comparing DCGANs in (a) and (c) and PGGANs in (b) and (d), PGGANs generate facial images that look more natural. Also, previous methods in (a) and (b) set the gender to neutral when inputted attributes are Eyeglasses+Smiling. Additionally, for Male+Goatee, previous methods reflect smiling in a few images. By contrast, our method in (c) and (d) is able to generate images that fulfill indicated conditions.

Moreover, Figure 5 shows Weighted Condition PGGANs can also generate natural high-quality facial images that fulfill condition and have higher resolution. Thus, Weighted Condition PGGANs can generate high-resolution images with clear facial details. Figure 6 shows facial images reconstructed by us-

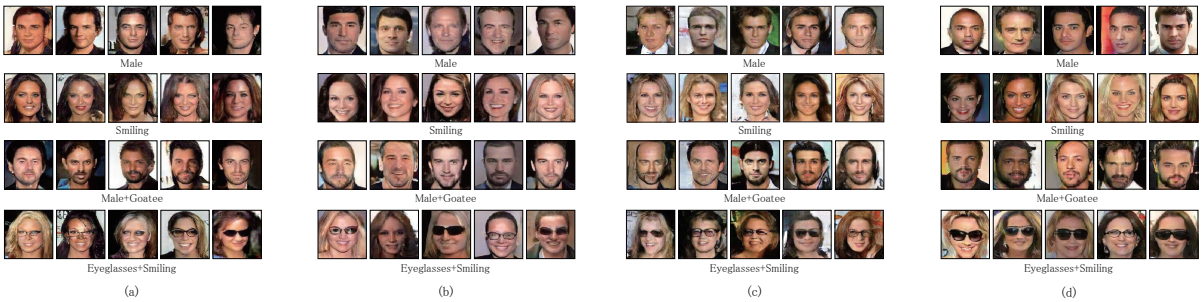

Figure 4: Facial images generated by different methods. Both (a) and (b) are generated by previous methods. (c) and (d) are generated by our method. (a) and (c) show results for DCGANs. (b) and (d) show results for PGGANs.

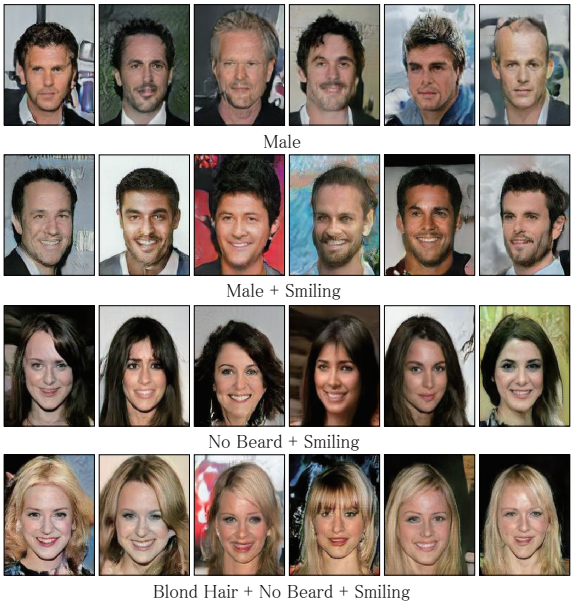

Figure 5: Facial images [192×256 pixels] generated by Weighted Condition PGGANs. Used facial conditions are Male, Blond Hair, Eyeglasses, No Beard, and Smiling.

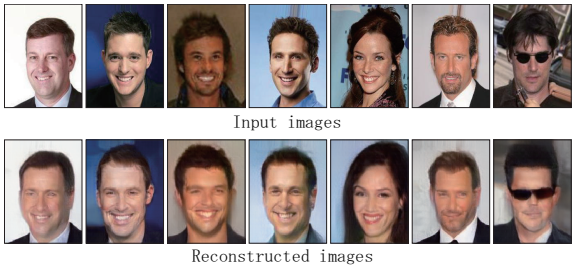

Figure 6: Input images and images reconstructed by our algorithm.

ing the proposed algorithm. In this image generation experiment, we input facial attributes given real images to the encoder and generator. Reconstructed images cannot completely maintain real images' identity but can maintain facial attributes, face direction, and background color. Therefore, we are able to argue that the generator of our proposed method can extract global features of images inputted to the encoder.

The results of evaluating the generated image quantitatively are shown in Table 1. Note that generated image are 128×128[pixels] in all methods. Weighted Condition DCGANs has a 0.03 lower Inception Score and 21.3 lower FID than Conditional DCGANs. In the subjective evaluation, Conditional DCGANs scores higher than our Weighted Condition DCGANs. Conditional PGGANs and our Weighted Condition PGGANs have similar Inception Scores, but our method has higher FID and subjective evaluation score. It is possible to confirm Inception Score is close to real images score when compare PGGANs and DCGANs. Both Weighted Condition PGGANs and DCGANs drastically reduce FID, especially PGGANs. Therefore, we argue that Weighted Condition PGGANs can improve the quality of generated images, but we think that some conditions vanish because the network of PGGANs is very deep. Thus, our proposed method effectively generates facial images that fulfil conditions by using a deep architecture network.

4.3 Effective Multi-Tasks and Weighted Conditions

To evaluate the effectiveness of introducing condition recognition to the discriminator and weighted conditions to the generator, we built two networks and then compared objective evaluation results of generated facial images. In the first network, Recognition Branch is removed from our discriminator, and in the second network, convolutional layers of conditions are removed from our generator. Table 2 shows evaluation results. According to results, Inception Score was about 0.1 lower and FID 10.0 higher when the discriminator did not have condition recognition. Moreover, the Inception Score was about 0.1 lower and FID 13.5 higher when the generator did not have weighted

Table 1: Evaluation results for various evaluation methods

Methods	Inception Score ↑	FID ↓	Subjective Evaluation (21 people) ↑
Real Images	1.97	-	-
Conditional DCGANs	1.70	402.4	53.1
Weighted Condition DCGANs (Proposed)	1.67	381.1	46.9
Conditional PGGANs	1.68	450.4	44.5
Weighted Condition PGGANs (Proposed)	1.73	387.6	55.5

Table 2: Comparison Inception Score and FID with and without conditions recognition and weighted conditions.

Recognized Conditions	Inception Score ↑	FID ↓	Weighted Conditions	Inception Score ↑	FID ↓
✓	1.73	387.6	✓	1.73	387.6
	1.62	397.6		1.65	401.1

conditions. Therefore, high quality and natural images can be generated by introducing weighted conditions to the generator and Recognition Branch to the discriminator.

4.4 Discussion

The reason the proposed method generated facial images that fulfilled indicated conditions is assumed to be that optimal facial attributes in each layer were reflected by weighted conditions. Therefore, we visualize the contribution of weighted conditions to the generator. Contribution rates are calculated with a weight filter in each convolutional layer. Contribution rate C_t is given as

$$C_t = \frac{1}{N}\sum_{n=1}^{N}\frac{|W_{t,n}|}{\sum_{m=1}^{M}|W_{m,n}|}, \tag{5}$$

where N, M, W and t are the number of filters, number of attributes, weight filter, and a target attribute, respectively. Figure 7 shows the contribution rates of Male, Blond Hair, Eyeglasses, No Beard, and Smiling to images generated at each resolution by Weighted Condition PGGANs. Blond Hair + No Beard + Smiling contribute more to middle images. Contribution rates of Male and Blond Hair are highest in low resolution and then tend to decrease as resolution becomes higher. The contribution rate of smiling increases from the input layer to hidden layers and then decreases toward the output layer. Furthermore, the contribution rates of Eyeglasses and No Beard are highest in high resolution images. Facial expressions are clearly generated after the layer where the contribution ratio of Smiling is the highest. Thus, in low resolution, global facial attributes have higher contribution rates, and in high resolution, detailed facial attributes have higher contribution rates, so our proposed method seems to be able to generate natural facial images that fulfill conditions.

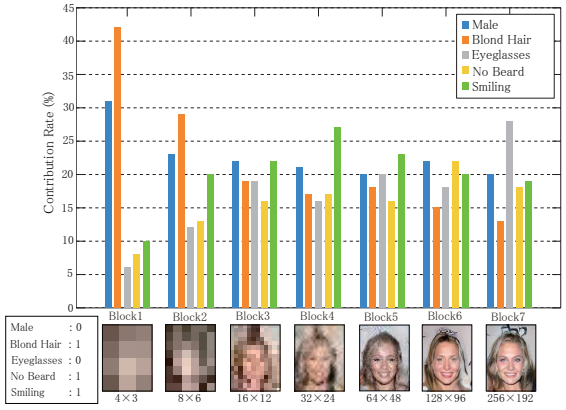

Figure 7: Contribution rate and generated image at each depth

5 Conclusions and Future Works

In this paper, we proposed a facial image generation method that introduces weighted conditions to both DCGANs and PGGANs and a new image reconstruction algorithm with an encoder. Condition can be stepwisely reflected by inputting weighted conditions. Moreover, conditions inputted to the generator can be easily reflected by a multi-task discriminator. The proposed method is able to generate facial images that fulfil conditions in both DCGANs and PGGANs. Evaluation results showed our method is able to drastically reduce the Fréchet Inception Distance (FID) score compared with previous methods. The encoder using our algorithm can obtain effective features in input image reconstruction. However, our algorithm has difficulty completely reconstructing input images. In future work, we will increase the resolution of the generated images and attempt to stabilize image generation.

REFERENCES

Augustus, O., Christopher, O., and Jonathon, S. (2017). Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning*.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Neural Information Processing Systems*.

Donahue, J., Philipp, K., and Darrell, T. (2017). Adversarial Feature Learning. In *International Conference on Learning Representation*.

Gauthier, J. (2014). Conditional generative adversarial nets for convolutional face generation. In *Convolutional Neural Networks for Visual Recognition*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Neural Information Processing Systems Conference*.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. In *Neural Information Processing Systems*.

Karras, T., Aila, T. A., Laine, S., and Lehtinen, J. (2018). Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representation*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representation*.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. In *arXiv*.

Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representation*.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. In *International Conference on Machine Learning*.

Rosca, M., Lakshminarayanan, B., Warde-Farley, D., and Mohamed, S. (2017). Variational Approaches for Auto-Encoding Generative Adversarial Networks. In *arXiv*.

Sergey, I. and Christian, S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*.

Yann, L., Leon, B., Yoshua, B., and Patrick, H. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*.

Zhang, Z., Song, Y., and Qi, H. (2017). Age progression/regression by conditional adversarial autoencoder. In *International Conference on Conputer Vison and Pattern Recognition*.

Ziwei, L., Ping, L., Xiaogang, W., and Xiaoou, T. (2015). Deep learning face attributes in the wild. In *International Conference on Computer Vision*.