applied sciences

MDPI

Article

Unsupervised Generation and Synthesis of Facial Images via an Auto-Encoder-Based Deep Generative Adversarial Network

Jeong gi Kwak [1] and Hanseok Ko [1,*]

School of Electrical Engineering, Korea University, Seoul 136-701, Korea; jgkwak@ispl.korea.ac.kr
* Correspondence: hsko@korea.ac.kr

Received: 26 December 2019; Accepted: 11 March 2020; Published: 14 March 2020

Abstract: The processing of facial images is an important task, because it is required for a large number of real-world applications. As deep-learning models evolve, they require a huge number of images for training. In reality, however, the number of images available is limited. Generative adversarial networks (GANs) have thus been utilized for database augmentation, but they suffer from unstable training, low visual quality, and a lack of diversity. In this paper, we propose an auto-encoder-based GAN with an enhanced network structure and training scheme for Database (DB) augmentation and image synthesis. Our generator and decoder are divided into two separate modules that each take input vectors for low-level and high-level features; these input vectors affect all layers within the generator and decoder. The effectiveness of the proposed method is demonstrated by comparing it with baseline methods. In addition, we introduce a new scheme that can combine two existing images without the need for extra networks based on the auto-encoder structure of the discriminator in our model. We add a novel double-constraint loss to make the encoded latent vectors equal to the input vectors.

Keywords: generative models; GAN (Generative adversarial networks); facial image; generation; database augmentation; synthesis

1. Introduction

In the last few years, deep neural networks (DNNs) have been successfully applied to a range of computer vision tasks, including classification [1–3], detection [4–6], segmentation [7,8], and information fusion [9,10]. However, because data augmentation is essential for the effective training of DNNs, and because there are numerous image-to-image translation and information fusion problems that need to be overcome, deep generative models have received significant attention. In this field, research on facial datasets has been particularly active, because they have a large number of real-world applications, such as facial classification and the opening of closed eyes in photos. Despite this increase in research interest, implementing generative models remains challenging because the process required to generate realistic images from low-level to high-level information is complex.

Since Goodfellow et al. [11] first proposed the generative adversarial network (GAN), which is based on adversarial learning between two networks, a generator and a discriminator, many GAN models have demonstrated excellent performance in terms of their photo-realistic output. The key principle underlying the use of a GAN is to ensure that the probability distribution of the generated data is close to that of the real data via the adversarial training of the generator and discriminator. In the early stages of training, the generator may generate poor-quality images; thus, the discriminator can easily distinguish between real and fake samples. As the generator learns more during training, its output becomes more photo-realistic and the discriminator finds it more difficult to distinguish

between real and fake samples. When the training reaches convergence, the generator can generate realistic but fake images. However, many GAN models suffer from instability during the training process, leading to problems such as mode collapse and the lack of diversity.

BEGAN [12] is an auto-encoder-based GAN model with an auto-encoder architecture as the discriminator. Unlike many existing GAN models [11,13,14] that attempt to directly match the real data distribution, this model seeks to match the loss distribution of the auto-encoder. The BEGAN developers introduced an equilibrium hyperparameter to maintain the balance between the generator and the discriminator. It makes it possible for a user to control the visual quality and diversity of an image by changing the parameters. However, it suffers from the trade-off between diversity and quality, is subject to mode collapse, and occasionally fails to generate high-quality images during the training phase.

StyleGAN [15] can generate photo-realistic output images using a style-based generator that considers the scale-specific characteristics of the generated image. Each layer in the StyleGAN generator consists of several convolutional layers and adaptive instance normalization (AdaIN) [16] layers. The AdaIN layers utilize latent vectors as input and then utilize their information with an affine transform. In addition, StyleGAN can perform style mixing, in which an image generated using two different latent vectors has both characteristics. However, StyleGAN generates an image from noise; thus, it cannot mix two existing images; i.e., it does not take existing images as input. Synthesizing two existing images using the model requires the training of an additional network that can encode real images into the latent space of StyleGAN.

Motivated by StyleGAN, we propose a generator that takes two latent vectors as input based on the scale-specific role of each layer in the generator. The front layers are involved in the creation of high-level features such as the overall shape of the face, while the back layers are involved in lower-level features such as hair color and the microstructure. Our discriminator is trained to reconstruct only real images, and its decoder has the same structure as the generator. This divided structure of the generator and decoder that utilize different latent vectors to assign scale-specific roles in image generation improves the visual quality of the image.

We also adopt a training technique that differs from that used in the conventional BEGAN model. The instability of GANs usually occurs when generating high-resolution images; thus, we adopt the progressive growth concept for the generator and discriminator introduced in [17]. The size of a generated image at the beginning of the training process is small, but it becomes twice the size after several epochs. This training scheme reduces instability and consequently improves the visual quality of the output images.

In addition to generating images using random vectors, we also propose a method to synthesize two existing images by exploiting the auto-encoder structure of our discriminator. The encoder of discriminator learns to encode both real and fake samples during the training process; thus, it does not need to train an additional model. However, in order for the decoder or generator to combine real images, the encoded latent space of the real images should be similar to that of the fake images. To guarantee this, we propose the novel double-constraint loss function, which constrains the latent vectors of encoded real images. Therefore, the images are combined when the decoder decodes an image using the latent vectors obtained from the different images in an unsupervised manner.

This paper is structured as follows. Section 2 presents the theoretical background and provides a detailed description of the proposed model. We then demonstrate the superiority of our model by qualitatively and quantitatively comparing it to conventional models [12,18] in Section 3. Concluding remarks are presented in Section 4.

2. Proposed Method

This section describes our proposed model in detail by first introducing the BEGAN baseline model with a brief explanation of the auto-encoder-based GAN and then outlining the structure of our

proposed model and its training strategy. Subsequently, we introduce a method for combining facial images using our model.

2.1. BEGAN Baseline Model

Conventional GANs have a generator and a discriminator; the generator creates fake images, whereas the discriminator receives both real and fake images as input and attempts to distinguish them. The goal of a GAN is to match the probability distribution of the fake samples generated by the generator to that of the real samples. Therefore, the output of the discriminator is essentially a probability score, and this is fed into the loss function. However, BEGAN has a discriminator with an auto-encoder structure, meaning that the output of the discriminator is an image of the same size as the input.

Auto-encoder-based GAN models can be optimized by reducing the Wasserstein distance between the reconstruction loss distributions of the real and fake images rather than their sample distributions directly [12,19]. The discriminator attempts to reconstruct only real images, but the generator attempts to produce an image that can be accurately reconstructed by the discriminator. Therefore, the reconstruction performance of the discriminator is crucial for the generator to be able to produce high-quality output. If the decoder within the discriminator produces poor-quality images when reconstructing the input, the generator could easily fool the discriminator with those poor-quality images.

Berthelot et al. [12] introduced the hyperparameter $\gamma \in [0, 1]$ to maintain the balance between generator and discriminator loss, defined as

$$\gamma = \frac{\mathbb{E}\left[\mathcal{L}(\mathrm{G(z)})\right]}{\mathbb{E}\left[\mathcal{L}(x)\right]}, \tag{1}$$

where $\mathcal{L}(\cdot)$ denotes the L_1 or L_2 reconstruction error from the auto-encoder; i.e., the discriminator. $\mathbb{E}(\cdot)$ denotes expectation operator. $G(z)$ denotes a fake image from the generator and x denotes a real image. This ratio (γ) enables users to control the balance between the visual quality and diversity of the output images. If γ is low, the model focuses more on reducing the reconstruction loss of the real images; i.e., the auto-encoding ability of the discriminator increases. This leads to higher visual quality and lower diversity. However, BEGAN has limitations in terms of visual quality and diversity due to the inherent structure of the generator, the lack of reconstruction ability in the discriminator, and unstable training.

2.2. The Proposed Model

2.2.1. Network Architecture

We propose the novel auto-encoder-based GAN architecture illustrated in Figure 1. Our generator takes two latent vectors and consists of several blocks, with each block handling a specific resolution. The latent vectors are fed into each block and transformed by the affine transformation layer. We use an AdaIN [16] layer that stylizes feature maps with information from the affine transformation layer. We divide the generator into front and back modules, with the front module generating feature maps of a relatively low resolution (32×32) and the back module generating the final output image. z_1 is fed into the front module, and z_2 is fed into the back module, meaning z_1 is associated with the overall structure of the image (e.g., the shape or appearance of the face), whereas z_2 is associated with the details of the image (e.g., the microcharacteristics of the face or hair color). The bottom of Figure 1 presents the details of each block. Initially, the input features are upscaled, and there are three sets of Conv-ELU-AdaIN layers. As mentioned above, the AdaIN layer normalizes the features and matches them to new statistics (i.e., the mean and variance from the affine transformation layer). AdaIN is formulated as

$$\mathrm{AdaIN}(x, y) = \sigma(\mathrm{y})\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y), \tag{2}$$

where x denotes the feature map, and the new mean and variance ($\sigma(y)$ and $\mu(y)$, respectively) are calculated by affine transformation with the input latent vectors. Because of the scale-specific role of each layer, the visual quality of the output images is improved.

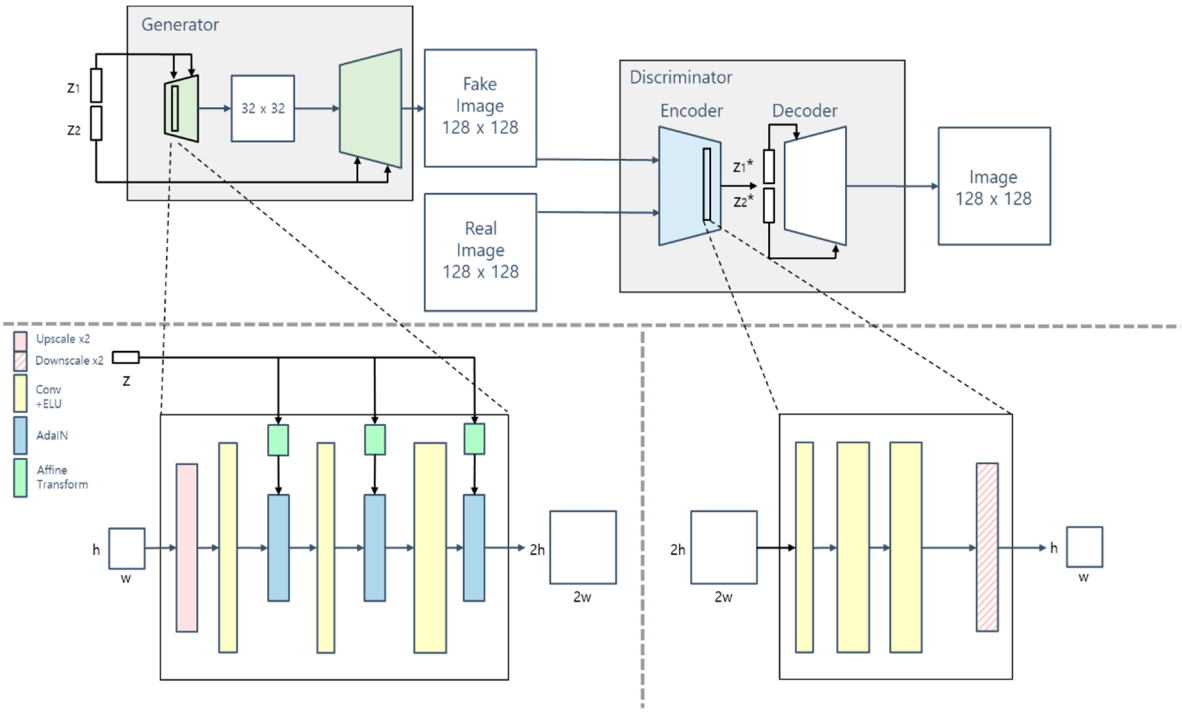

Figure 1. Overview of the proposed model, consisting of two networks; generator and discriminator. The generator takes two input vectors and generates a fake image. The discriminator takes a real or fake image and it is learned to reconstruct only real sample (Top). There are AdaIN layers which stylize feature maps with transformed input vector after convolutional layer in each block of the generator (Left bottom). The encoder down-samples input image to two latent vectors with convolutional layers and down sampling layers (Right bottom).

The discriminator of our model has an auto-encoder structure that consists of an encoder and a decoder. The encoder takes a real or fake image as input and encodes it as two latent vectors z_1^* and z_2^* of the same size as z_1 and z_2 respectively. The decoder then decodes the image with z_1^* and z_2^*. Because the decoder has the same structure as the generator, z_1^* and z_2^* affect different scale-specific characteristics.

2.2.2. Objective Function

A fake image generated from the input vector $z_1,\ z_2$ can be expressed as $G(z_1,\ z_2)$. The goal of the discriminator is to distinguish real image x from fake image $G(z_1,\ z_2)$. Therefore, the discriminator attempts to reconstruct x only, not $G(z_1,\ z_2)$. On the other hand, the generator attempts to produce an image that can be reconstructed well by the discriminator. As a result of the adversarial training of the generator and the discriminator, the output images from the generator become more realistic to deceive the discriminator. In other words, the generator is trained to reduce the Wasserstein distance between the loss distributions of real and fake samples in the auto-encoder. The adversarial loss of the generator and discriminator can be expressed as

$$\begin{aligned} L_D &= L(x;\theta_D) - k_t L(G(z_D;\theta_G);\theta_D), \\ k_{t+1} &= k_t - \lambda_k(\gamma L(x) - L(G(z_G))\ \textit{for each step } t, \end{aligned} \tag{3}$$

and

$$L_G = L(G(z_D;\theta_G);\theta_D), \tag{4}$$

where $L(\cdot)$ denotes the L_1 loss from the auto-encoder, and k_t is the parameter that controls the proportion of generator and discriminator loss introduced in BEGAN. This is required because the discriminator cannot achieve a suitable reconstruction quality at the beginning of training. At this time, k_t has a value close to zero and gradually increases as training progresses.

As mentioned in Section 2.2.1, z_1 and z_2 are both involved in the generation of different scale-specific areas. To apply this principle to the decoder, we add a novel constraint on the encoded latent vectors, referred to as double-constraint loss. It includes the difference between the input vector and the encoded vector as defined by

$$\begin{aligned} L_{dc} &= \|z_1 - z_1^*\|_1 + \|z_2 - z_2^*\|_1, \\ [z_1^*\ z_2^*] &= Enc(G(z_1,\ z_2)), \end{aligned} \tag{5}$$

where $Enc(\cdot)$ denotes the output of the encoder. The double-constraint loss is designed to stabilize training because the inputs of the generator and decoder would be similar. It can also be extended to the synthesis of existing images because real samples are mapped to a space similar to the latent space of the input. Hence, the generator loss can be modified as

$$L_G = L(G(z_D; \theta_G); \theta_D) + \alpha \cdot L_{dc}, \tag{6}$$

where hyperparameter α represents a weighting factor for the double-constraint loss.

2.2.3. Training Scheme

Unstable training is a major concern when using GANs, and it can occasionally result in mode collapse or low-quality output. In auto-encoder-based GAN models in particular, the reconstruction performance of the discriminator is a decisive factor in establishing the visual quality of an output image. However, excellent reconstruction performance cannot be guaranteed because the importance of the reconstruction error for a real image decreases as k_t increases, as can be seen in Equation (3). Training a discriminator on relatively large images (e.g., 128×128) is slow and difficult, and k_t becomes larger because the discriminator does not effectively function as an auto-encoder. Motivated by PGGAN [17], our model attempts to overcome this problem by starting the training process with low-resolution images. In other words, the size of the training images increases as training progresses (Figure 2). When the size of an image is larger, new layers are added to both the generator and discriminator to adjust the size of the input and output correctly. While PGGAN starts training with 4×4 images, our model begins with 32×32 images because our model performs sufficiently without threatening stability when generating images with sizes of 32×32 or lower, thereby reducing the training time. After a few epochs of training, the size of the training images is doubled, and new layers are added to the generator, encoder, and decoder while maintaining the weights in the conventional layers. By progressively training the generator and the discriminator in this manner, our model achieves better reconstruction performance than when k_t remains constant. Because the discriminator is trained to some extent in the previous stage, the training process becomes more stable and the visual quality is higher than when training with 128×128 images directly. In addition, the layers of the generator and the decoder can accurately reflect the spatial properties of their input.

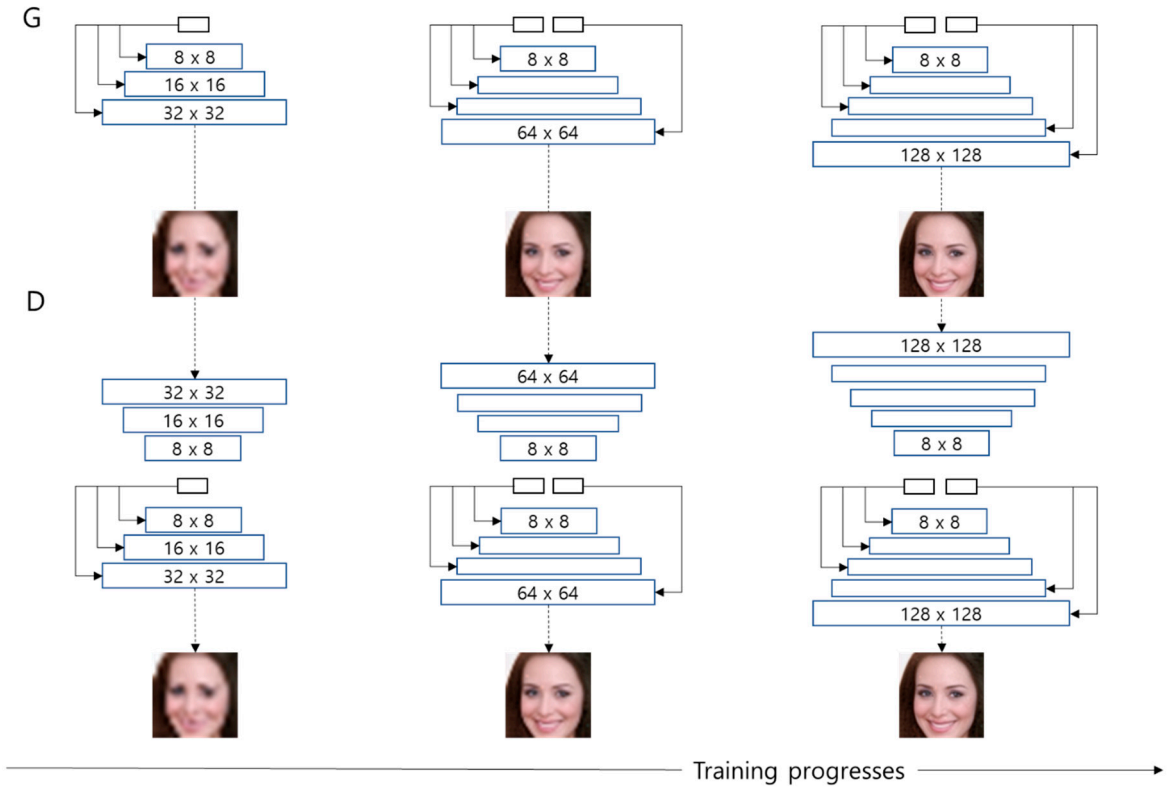

Figure 2. Progressive training of the proposed generator and discriminator. Our model starts with a 32×32 image in the first stage, and the size of the training images is doubled in the next stage.

2.2.4. Facial Synthesis Method

In addition to generating images from random noise, as with other unsupervised GANs, our model can also be used to synthesize two images. StyleGAN introduced style mixing, which exploits two or more input vectors, but it was used on only random noise input, not existing images. To mix two existing images, an additional encoder needs to be trained to map the images onto the latent space of the input. However, our model does not require an additional network because our discriminator already has an encoder. By taking advantage of the auto-encoder structure of our discriminator, we present a method for mixing existing images. The encoder encodes an input image as two latent vectors, and they are exploited in different layers of the decoder. Reconstruction occurs when the decoder uses the two latent vectors from a single image. However, if the decoder exploits a combination of the two latent vectors from two different images, the output of the decoder is a mixed image.

Let X and Y denote the two images to be mixed; the output of the encoder when the input is X and Y can be expressed as

$$[z^*_{X_1}\ z^*_{X_2}] = Enc(\mathrm{X}),\ \left[z^*_{Y_1}\ z^*_{Y_2}\right] = Enc(Y). \tag{7}$$

If the decoder decodes the image using $z^*_{X_1}$ and $z^*_{X_2}$, it reconstructs X, and if it uses $z^*_{Y_1}$ and $z^*_{Y_2}$, it reconstructs Y; i.e.,

$$X^* = Dec\left(z^*_{X_1},\ z^*_{X_2}\right) \rightarrow Reconstruction\ of\ X, \tag{8}$$

$$Y^* = Dec\left(z^*_{Y_1},\ z^*_{Y_2}\right) \rightarrow Reconstruction\ of\ Y, \tag{9}$$

where X^* and Y^* denote the reconstructed images of X and Y, respectively, and $Dec(.)$ denotes our decoder. To synthesize X and Y, the decoder needs to take the latent vectors from the two images as input. A mixed image of X and Y is acquired by exploiting a combination of the latent vectors from the two images (e.g., $z^*_{X_1}$ and $z^*_{Y_2}$), as illustrated in Figure 3. The two blue boxes in Figure 3 represent the two parts of the decoder; i.e., one is involved in generating a 32×32 feature map from a given latent

vector, and the other is involved in generating a 128 × 128 image from a given 32 × 32 feature map. Therefore, the synthesis process can be expressed as

$$I_{X,Y} = Dec\left(z^*_{X_1}, z^*_{Y_2}\right) \rightarrow Synthesis\ of\ X\ and\ Y, \tag{10}$$

where $I_{X,Y}$ is an image that has the structural or coarse-scale characteristics of X and the details or fine-scale characteristics of Y.

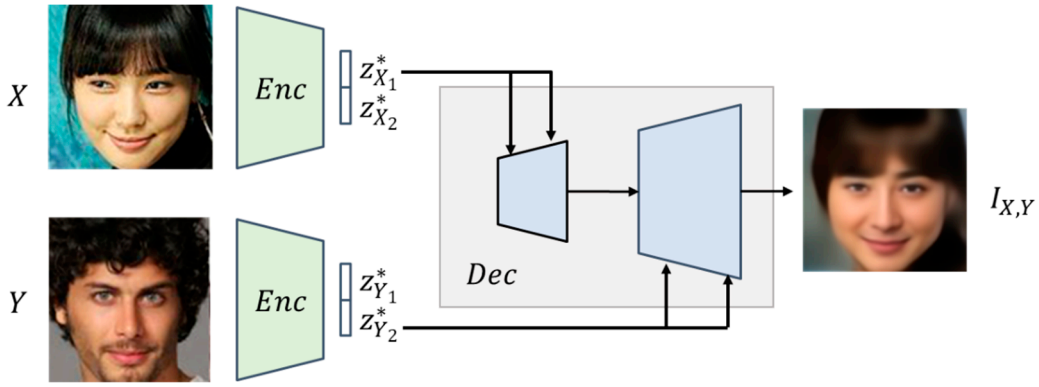

Figure 3. The facial image synthesis process for our model. The decoder takes one encoded vector from each image.

3. Experimental Results

In this section, we first explain our experimental setup and then present qualitative and quantitative comparisons of the performance of our model with those of other auto-encoder-based models.

3.1. Experimental Setup

We used the CelebFaces Attributes (CelebA) dataset (Figure 4) [20], which consists of 202,599 facial images of celebrities cropped to 178 × 218. We cropped each image further to 170 × 170, and then resized them to 128 × 128. For progressive training, we downsampled the images to 32 × 32 and used them in the first stage. The height and width of each training image were doubled every five training epochs. In our experiments, the coefficients of the objective functions in Equations (3) and (6) were set to $\gamma = 0.5$ and $\alpha = 0.1$. We used the ADAM [21] solver with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and the learning rate was set initially to 0.0005. L_1 loss was adopted as the loss function for the auto-encoder. All other parameters were the same as in BEGAN. We used Tensorflow with cuDNN as the deep-learning framework and an NVIDIA GTX 1080Ti graphics card.

Figure 4. CelebFaces Attributes (CelebA) dataset.

3.2. Qualitative Results

We conducted qualitative analysis by comparing the output of our model with those of two other auto-encoder-based GAN models, BEGAN [12] and BEGAN-CS [18]. BEGAN-CS adds a latent constraint to BEGAN. The results are shown in Figure 5. The columns (a) to (c) represent 5, 10, and 15 epochs, respectively, while each row represents the qualitative results from the compared methods. The output images are produced by the generator and the input vectors are sampled randomly from a Gaussian distribution. It should be noted that the output of our model in (a) (5 epochs) has a lower resolution than the other models because of the progressive learning strategy it employs. The visual quality of the images improves as training progresses in all three models. However, the results from BEGAN contain some artifacts, such as checkerboard patterns, while BEGAN-CS produces blurred and unstructured facial images (Figure 5d). Once the size of the training images is increased, our model produces similar visual quality in Figure 5b and clearer images than the other models after 15 epochs (Figure 5c,d).

Figure 5. Qualitative results for facial image generation. The rows from top to bottom present the results of BEGAN, BEGAN-CS, and the proposed model, respectively. Each column represents five epochs; i.e., (**a**) epoch 5, (**b**) epoch 10, and (**c**) epoch 15. (**d**) An enlargement of the red box in (c).

3.3. Quantitative Results

It is difficult to verify the diversity of output images using several images. Therefore, we conducted quantitative experiments using the Fréchet inception distance (FID) [22]. The FID score can be used to measure the quality and diversity of images. The FID score is calculated using Equation (11):

$$\mathrm{FID} = \|\mu_x - \mu_y\|_2^2 + Tr\left(\Sigma_x + \Sigma_y - 2\left(\Sigma_x\Sigma_y\right)^{\frac{1}{2}}\right), \tag{11}$$

where x and y denote the image sets x and y. In our experiments, x and y consist of real images and fake images, respectively. Because the FID score considers the mean (μ) and variance (Σ) of the images, it can represent the visual quality and diversity of the images. If the two sets of images have a similar probability distribution, the FID score is low (Equation (11)). Therefore, lower FID scores are better

when comparing GAN models. We measured the FID score based on 5000 real samples and 5000 fake samples in epoch 15 for each model. The results are summarized in Table 1.

Table 1. Visual quality in terms of the Fréchet inception distance (FID) score, where a lower score is better.

	BEGAN	BEGAN-CS	Style-AEGAN (ours)
FID	47.93 ± 1.18	50.31 ± 1.01	41.88 ± 1.08

Our model produces the best results. As a result, our model can be seen as superior in terms of image quality and diversity.

3.4. Facial Synthesis Results

We test the synthesis of facial images using our model as described in Section 3.4. In Figure 6, the right-most image in each row represents the synthesis output of the two left-side images. The front module of our decoder takes a latent vector encoded from the left image, and the back module takes a latent vector from the right image. The output image has the characteristics of both images but different scale-specific features. In other words, the output has the coarse-scale characteristics of the first image (e.g., the overall structures or locations of facial attributes) and the fine-scale features of the second image (e.g., the eyes or the skin color). Note that facial synthesis is achieved without requiring additional information, such as binary attribute labels for each image.

Figure 6. Qualitative results of facial synthesis with the proposed model. The right-most image is the synthesis of the two left-side images.

4. Conclusions

In this paper, we proposed an enhanced GAN model for unsupervised facial image generation and synthesis. To overcome the limitations of GAN models (particularly auto-encoder-based models), we first introduced an enhanced generator and discriminator structure. Our generator and decoder utilize two input vectors, and every block reflects the information from the input vectors with adaptive instance normalization layers. Each layer plays a role in producing scale-specific components of the facial image. We also applied a progressive learning method to the proposed auto-encoder-based model, in which the training process was divided into several stages depending on the size of the training image. Consequently, our model can both generate and synthesize facial images via an auto-encoder

structure. Our model can generate arbitrary images because it also takes noise as input and synthesizes two existing images using an encoder and decoder within the discriminator. Therefore, it does not require additional training to encode existing images or a pre-trained network. We demonstrated that the visual quality and diversity of the output images were higher than those of the baseline models using both qualitative and quantitative analysis. Additionally, we presented a method for synthesizing two existing images by exploiting the auto-encoder structure of the discriminator. Our model did not need to train a subnetwork that could encode the images for mixing. All of the networks in our model were trained in an end-to-end manner without the labeling of the images. In future research, we will further investigate this novel method from a variety of perspectives to enhance the visual quality of the output images and to ensure stable training for large-scale image generation. Furthermore, we will extend our model for use in not only unsupervised generation tasks but also conditional image generation or synthesis tasks, such as image-to-image translation.

Author Contributions: Conceptualization: J.g.K.; methodology: J.g.K.; software: J.g.K.; investigation: J.g.K.; writing—original draft preparation: J.g.K.; writing—review and editing: H.K.; supervision: H.K. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This research was supported by a National Research Foundation (NRF) grant funded by the MSIP of Korea (number 2019R1A2C2009480).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNET classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Lake Tahoe, CA, USA, 2012; pp. 1097–1105.
2. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Montreal, QC, Canada, 2015; pp. 91–99.
5. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Boston, MA, USA, 7–12 June 2015; pp. 1440–1448.
6. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
7. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
8. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
9. Chen, Z.; Li, W. Multisensor feature fusion for bearing fault diagnosis using sparse auto-encoder and deep belief network. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 1693–1702. [CrossRef]
10. Liu, Y.; Chen, X.; Peng, H.; Wang, Z. Multi-focus image fusion with a deep convolutional neural network. *Inf. Fusion* **2017**, *36*, 191–207. [CrossRef]
11. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Montreal, QC, Canada, 2014; pp. 2672–2680.

12. Berthelot, D.; Schumm, T.; Metz, L. Began: Boundary equilibrium generative adversarial networks. *arXiv* **2017**, arXiv:1703.10717.
13. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Paul Smolley, S. Least squares generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 2794–2802.
14. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
15. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4401–4410.
16. Huang, X.; Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 1501–1510.
17. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. *arXiv* **2017**, arXiv:1710.10196.
18. Chang, C.C.; Hubert Lin, C.; Lee, C.R.; Juan, D.C.; Wei, W.; Chen, H.T. Escaping from collapsing modes in a constrained space. In Proceedings of the European Conference on Computer Vision, Salt Lake City, UT, USA, 18–23 June 2018; pp. 204–219.
19. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Long Beach, CA, USA, 2017; pp. 5767–5777.
20. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision, Boston, MA, USA, 7–12 June 2015; pp. 3730–3738.
21. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
22. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Long Beach, CA, USA, 2017; pp. 6626–6637.