eISSN: 2976-1190

Journal of Engineering and Sciences

Vol. 2 Issue 1 November 30, 2023

Journal homepage: ***Journal of Engineering and Sciences (nepjol.info)***

Publisher: Research Management Cell, Pashchimanchal Campus, Lamachour, Pokhara

Facial Attribute Editing Using Generative Adversarial Network

Mukunda Upadhyay[1], Badri Raj Lamichhane[1], Bal Krishna Nyaupane[1]

[1]*Department of Electronics and computer Engineering, IOE, Pashchimanchal Campus, Tribhuvan University, Nepal*

(Manuscript Received:27/08/2023; Revised:21/11/2023/; Accepted:23/11/ 2023)

Abstract

Facial attribute editing tasks have immense applications in today's digital world, including virtual makeup, generating faces in the animation and gaming industry, social media face image enhancement and improving face recognition systems. This task can be achieved manually or automatically. Manual facial attribute editing, performed with software such as Adobe Photoshop, is a tedious and time-consuming process that requires an expert person. However, Automatic facial attribute editing tasks that can perform facial attribute editing within a few seconds are achievable using encoder-decoder and deep learning-based generative models, such as conditional Generative Adversarial Networks. In our work, we use different attribute vectors as conditional information to generate desired target images, and encoder-decoder structures incorporate feature transfer units to choose and alter encoder-based features. Later, these encoder features are concatenated with the decoder feature to strengthen the attribute editing ability of the model. For this research, we apply reconstruction loss to preserve other details of a face image except target attributes. Adversarial loss is employed for visually realistic editing and attribute manipulation loss is employed to ensure that the generated image possesses the correct attributes. Furthermore, we adopt the WGAN-GP loss function type to improve training stability and reduce the mode collapse problem that often occurs in GAN. Experiments on the Celebi dataset show that this method produces visually realistic facial attribute edited images with PSNR/SSIM 31.7/0.95 and 89.23 % of average attribute editing accuracy for 13 facial attributes including Bangs, Mustache, Bald, Bushy Eyebrows, Blond Hair, Eyeglasses, Black Hair, Brown Hair, Mouth Slightly Open, Male, No Beard, pale Skin and Young.

Keywords: Adversarial Learning; CGAN; Difference attribute vector; Facial attribute Editing; Feature transfer unit

1. Introduction

Facial attribute editing involves altering specific features of a facial image while preserving the remaining attributes and the person's identity. For instance, this could entail modifying hair color or introducing eyeglasses into a facial image while ensuring all other characteristics remain unaltered. This editing task has many real-life applications, including data augmentation, Virtual makeup and Facial image processing, social media image enhancement, automatic face recognition systems, and the game and animation industry. Facial attribute editing can be performed manually using complex software like Photoshop; however, achieving high-quality, realistic results in facial attribute editing requires skilled manpower, complex software, and sufficient time.

Deep learning [1] is a popular topic at present time due to its high accuracy for discriminative-focused tasks such as prediction and classification. These deep learning models are trained on extensive labeled data sets to achieve high accuracy. However, collecting large amounts of labeled data is impractical in certain situations, such as obtaining images of the same person in both female and male versions. One effective solution to this challenge is to utilize generative model [2] techniques, which learn to generate new data similar to the data it was trained on. Variational Auto Encoder [3] and GAN [4] are two notable examples of generative models, but the Variational Autoencoder uses pixel-wise reconstruction error (e.g., mean square error) as a loss function, which makes the model non-translation invariant and causes the output images to look blurry [5]. GAN consists of two networks, a generator and a discriminator, with different functions. The generator tries to generate more realistic data that can fool the discriminator. Whereas the discriminator tries to distinguish between the input (real) data and generator-generated (fake) data as correctly as possible. GANs can generate new images that resemble input data distribution, but they do not give us control over the generated images. That's why we adopt conditional GAN [6], which allows for generating output data based on specific input conditions and provides greater control over the output data. Conditional information enhances the generator's ability to produce specific outputs. In facial attribute editing, we utilize a conditional

[1]Corresponding author. Tel.: +977- 9801793799,
E-mail address: updmuku24@gmail.com

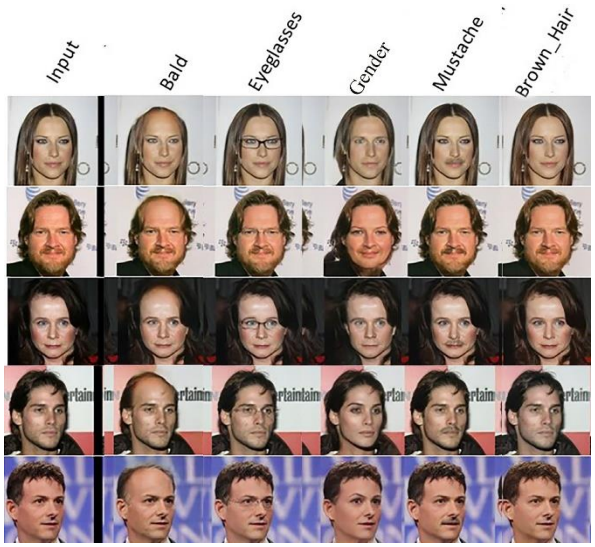

Figure 1: Facial attribute editing results from our proposed model

GAN to ensure output alignment with the provided conditional information.

Existing methods [7, 8, 9, 10 and 11] Train separate models for each attribute. Therefore, one has to develop different models for different attribute manipulation. H-GAN [10] is a combined version of VAE and GAN that manipulates the hairstyle of the source image. DNA-GAN [11] makes crossbreed images by swapping relevant latent blocks between images. Kim et al. [12] define attributes by different blocks of code and perform multiple attribute swapping between two inputs. Both Kim et al. and DNA-GAN are extended versions of GeneGAN [9]. IcGAN [5] compresses the input image into latent representations and decodes it with conditional information. However, this model trains the encoder and cGAN separately, limiting the reconstruction ability and flexibility of the generator [13]. Fader Networks [14] train encoder-decoder architecture such that the latent representation is free from salient information of the input image. Making latent representations free from facial attributes results in information loss and reduced reconstruction ability. StarGAN [15] performs image-to-image translation between multiple domains to achieve facial attribute editing, but it lacks fine-grained control over specific attribute editing. AttGAN [13] introduces three types of losses to train the model and uses a skip connection, similar to U-net [16], to transfer the encoder feature to the decoder, but the use of a skip connection reduces the attribute manipulation ability of the model [17]. Additionally, most existing methods use full target attribute vectors as conditional information. This often causes problems by manipulating other source attributes that must be kept constant. Only the attributes that require modification should be taken into account in order to preserve other details of the source image. Considering the above issues into account, we propose a novel model that can perform high-quality realistic facial attribute editing tasks, preserving all other attributes of the source image except the target attribute by utilizing a difference attribute vector as conditional information instead of a full target attribute vector.

2. Materials and Method

2.1 Methodology

GAN models are able to produce satisfactory results, but the Training of GAN is often unstable because the training process involves min-max games between two networks. So, the training process may fail to converge in scenarios where the generator excels at deceiving the discriminator or when the discriminator excels at precise classification. Additionally, the mode collapse phenomenon occurs when the generator learns to produce a limited number of variations of the data rather than being able to generate a wide variety of examples. We adopt the WGAN-GP [18] adversarial loss function to overcome these two issues, which provides greater stability and is less sensitive to model parameters. Importantly, the Wasserstein distance used to calculate the difference between the probability distribution of real and fake images is continuous and differentiable and continues to provide a linear gradient, even after the critic is well-trained [19]. In WGAN-GP, the discriminator architecture is called the critic due to the omission of a sigmoid activation function that confines outputs to binary values of 0 or 1, indicating real or fake samples. This modification allows the critic network to yield values across a broader range, enabling them to function as versatile critics. Instead of categorizing images strictly as real

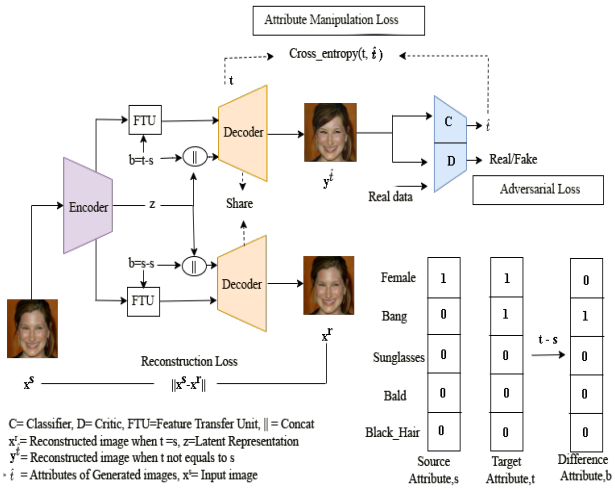

Figure 2: Block diagram of proposed model

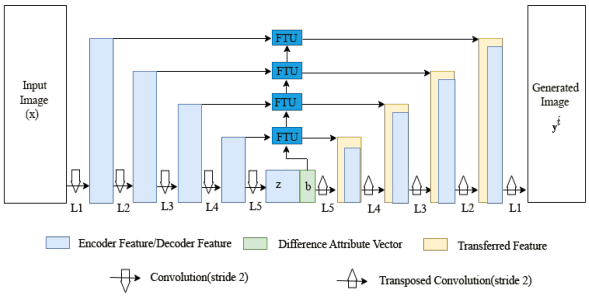

Figure 3: Detail structure of generator

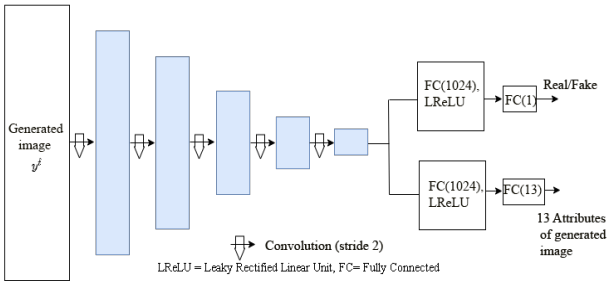

Figure 4: Detail structure of discriminator

or fake, the critic model scores the realness and fakeness of given image. The adversarial Process in WGAN-GP is expressed as:

$$\min_G \max_{\|D\|\leq 1} \mathrm{E}_{\mathbf{x}\sim p_r} D[(\mathbf{x})] - \mathrm{E}_{z\sim p(z)} D[G(\mathbf{z})] + G.P \quad (1)$$

Here, D denotes critic, constrained to be a 1-Lipschitz function. The first segment of the equation represents the output from the critic for real data 'x', The middle segment represents the critic output for fake data 'G(z)' and G.P represents the gradient penalty which helps to maintain the L2 norm of the critic gradients close to 1 offering easier optimization and convergence of model compared to the other GAN training methods[18]. The critic endeavors to enhance the gap between the real and generated data, as it intends to differentiate them precisely. Conversely, the generator network strives to lessen the difference between the real and the generated data, as it seeks to create data that is as real as possible. The discriminator adversarial loss, known as the critic loss in the context of WGAN-GP, is intricately linked to the quality of the images generated by the generator. This suggests that an improvement in the quality of the generated images corresponds to a reduction in the critic loss, moving it towards zero [19]. The block diagram of the proposed model is shown in Figure 2. The encoder, decoder, and FTU [20] make the Generator module, and Critic D and Classifier C make the Discriminator module.

2.1.1 Generator

A face image x^s with s binary source attributes represented by $s = [s_1, s_2, \ldots, s_n]$ is passed through a series of convolution layers of the encoder to find features and patterns present in the image. The output from l^{th} encoder layer is given by:

$$f^l = G^l_{enc}(x^s) \quad (2)$$

Here, G_{enc} denotes the output from the encoder, and the output from the last layer of this sub-module is given by

$$z = \{f^1, f^2, \ldots, f^5\} \quad (3)$$

The output from the fifth layer of the encoder is concatenated with a different attribute vector and directly passed to the decoder. To reuse the feature extracted by the earlier layer of the encoder, instead of a direct skip connection, features are transferred via FTU, which refines the encoder feature based on the provided difference attribute vector and then passes to the designated decoder layer, as shown in Figure 3. The decoder regenerates the image based on information received from FTU and z conditioned on b.

$$\hat{y^t} = G_{dec}(G_{enc}(x^s), b) \quad (4)$$

Here is the attribute edited image generated by the encoder and expected to possess the target attribute t.

2.1.2 Discriminator

Table 1 : Model component architecture

L	Encoder	Decoder	Critic	Classifier
1	Conv(64,4,2),BN,LReLU	DeConv(3,4,2)	Conv(64,4,2),IN,LReLU	
2	Conv(128,4,2),BN,LReLU	DeConv(128,4,2),BN,ReLU	Conv(128,4,2),IN,LReLU	
3	Conv(256,4,2),BN,LReLU	DeConv(256,4,2),BN,ReLU	Conv(256,4,2),IN,LReLU	
4	Conv(512,4,2),BN,LReLU	DeConv(512,4,2),BN,ReLU	Conv(512,4,2),IN,LReLU	
5	Conv(1024,4,2),BN,LReLU	DeConv(1024,4,2),BN,ReLU	Conv(1024,4,2),IN,LReLU	
6			FC(1024),LReLU	FC(1024),LReLU
7			FC(1)	FC(13),Sigmoid

Conv (d, k, s) and DeConv (d, k, s) represent convolution and Transposed convolution operation with a dimension 'd', kernel size 'k' and stride value's'. BN and IN denote Batch Normalization and Instant Normalization, respectively. 'L' represents Layer number of the encoder and decoder.

The images created by the generator are sent to the discriminator module to determine whether the generated image is accurate and realistic. The image is deemed accurate if it correctly possesses the target attributes. This task is performed by the classifier by comparing the attributes of the generated image with the target attributes. Realism is assessed by the critic through an adversarial process. The detailed architecture of the critic and classifier is shown in Figure 4.

2.2 Loss Function

The discriminator model is trained directly on both real and generator generated images, while the generator does not undergo direct training; instead, its training occurs through interaction with the discriminator model. The discriminator learns to furnish the loss function for the generator, and these two models enhance their performance concurrently. The implemented loss functions in this study include reconstruction loss, WGAN-GP adversarial loss, and attribute manipulation loss.

2.2.1 Reconstruction loss

This loss represents the L1 norm distance metric, which measures the distance between the reconstructed and the original image and is employed to maintain the integrity of the remaining attributes of the facial image. The resulting reconstructed image is expected to be identical to the input when the source and target attribute vector is the same, causing the difference attribute vector to be zero. This loss is formulated as follows:

$$L_{rec} = \| x^s - G_{dec}(x^s, b) \|_1 \quad (5)$$

Here, $G_{dec}(x^s,b)$ represents the output from the decoder given x^s and b as inputs, and $\| \|_1$ represents the L1 norm.

2.2.2 Adversarial Loss

This loss is implemented to guarantee the realism and indistinguishability of the generated images. The generator adversarial loss incentivizes the generator to create outputs resembling real data, while the critic adversarial loss quantifies the discriminator's ability to differentiate between real and generated data. The adversarial loss for Critic and Generator is formulated as follows:

$$\min_D L_{adv_D} = -\mathrm{E}_{x^s} D_{adv}(x^s) + \mathrm{E}_{y^{\hat{}}} D_{adv}(y^{\hat{}}) - \lambda \mathrm{E}_{\hat{x}}[(\| \nabla_{y^{\hat{}}} D_{adv}(\hat{x}) \|_2 - 1)^2] \quad (6)$$

$$\min_G L_{adv_G} = -\mathrm{E}_{x^s,b}[D_{adv}(y^{\hat{}})] \quad (7)$$

Here, $\mathbb{E}$ represents expectation, λ controls the strength of the gradient penalty term, and $x^{\hat{}}$ is sampled along pathways connecting pairs of real and generated images.

Table 2: Hyperparameters Configurations

S.N	Parameter Name	Value
1	Image size	128*128
2	Number of Critic update per generator update	5
3	Epoch	60
4	Optimizer	Adam (β_1=0.5, β_2=0.99)[21]
5	Learning rate	2E-4
6	Batch size	32
7	λ1, λ2	100, 10

2.2.3Attribute Manipulation Loss

This loss ensures that the generated images possess the desired target attributes. It is calculated using the cross-entropy loss between the predicted and the target attribute label. This loss is formulated as,

$$\min_G L_{cls_G} = \sum_{i=1}^{n} [t_i \log c_i(y^{\hat{}}) - (1 - t_i) \log(1 - c_i(y^{\hat{}}))] \quad (8)$$

$$\min_C L_{cls_C} = \sum_{i=1}^{n} [s_i \log c_i(x^s) - (1 - s_i) \log(1 - c_i(x^s))] \quad (9)$$

Here, c_i represents the prediction of i^{th} attribute, s^i and t^i represent i^{th} source and target attribute, respectively.

Considering the mentioned losses, the aim of training these two sub-modules can be framed as:

$$\min_G L_G = \lambda_1 L_{rec} + \lambda_2 L_{cls_G} + L_{adv_G} \quad (10)$$

$$\min_{D,C} = L_{cls_C} + L_{adv_D} \quad (11)$$

2.3 Dataset collection

The study employs the CelebA[22] dataset, which contains 202,599 aligned face images at 178 x 218 pixels, each annotated with 40 binary attributes indicating features like Bald, Bang, 5'o clock shadow, Attractive, and chubby. The first 28,000 images are used for model training, the next 1,000 for validation, and the remaining for testing. This experiment focuses on 13 distinct facial attributes, such as Bald, Bangs, Blond Hair, Black Hair, Brown Hair, Eyeglasses, Bushy eyebrows, Male, Mouth Slightly Open, No Beard, Mustache, Pale skin and Age as these attributes are highly discernible and widely used in existing literature. Images are resized to 128x128 pixels and then scaled to [-1,1] before applying the ReLU activation function.

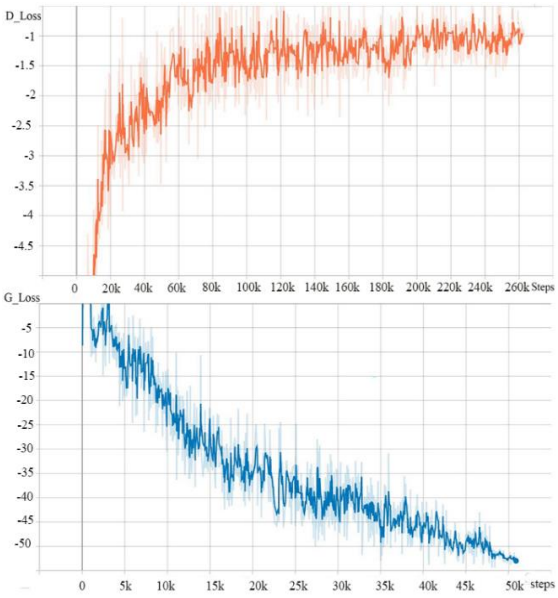

Figure 5: Critic and Generator adversarial loss vs. steps

This scaling aids in training stability and overall performance of deep neural networks [23].

2.4 Training Details

The detailed Model architecture is shown in Table 1, and Hyperparameters with their values are listed in Table 2. The tools and software's used in this thesis work are listed below:

- Python
- Tensorflow
- Numpy, pandas, PIL,Scikit-image
- Google Colab GPU with 32GB RAM.

3. Results and Discussion

The adversarial losses of the critic and generator throughout the training period are depicted in Figure 5. The curves in the figure illustrate that the critic loss value moves toward zero as the training steps increase. In the case of the generator, a larger score for fake images by the critic will result in a smaller generator adversarial loss, which encourages the critic to output a larger score for fake images because generator adversarial loss is equal to the negative of mean critic score on fake images [19] which is smaller and so on.

At the beginning of training, the model is just starting to learn the input data distribution, so the output of the first epoch is blurry. However, as training steps increase, the model gradually learns to reconstruct and edit attributes of the input image based on conditional information, as shown in Figure 6. Not only can the model manipulate a single attribute, but

Figure 6: Intermediate results of the model

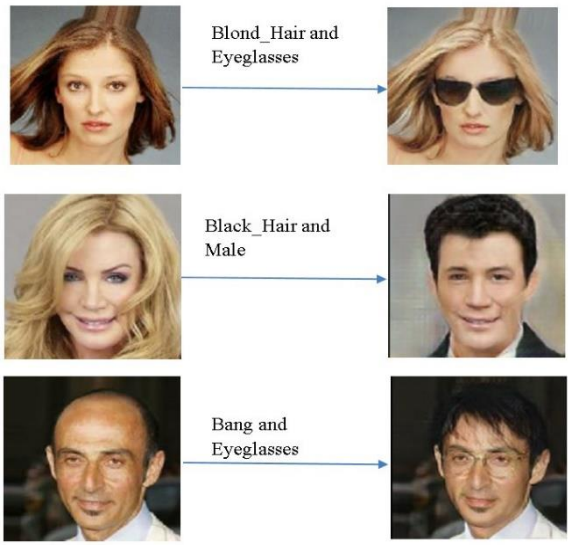

Figure 7: Multiple attribute editing

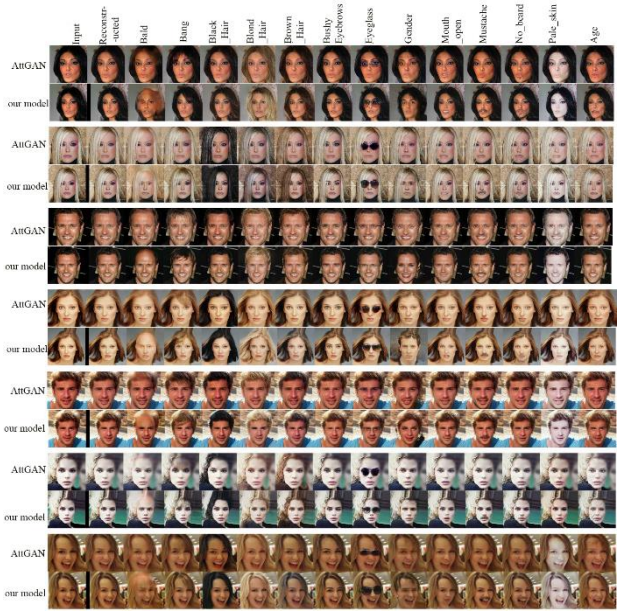

Figure 8: Comparison with AttGAN [13]

it is also capable of manipulating multiple attributes simultaneously, as shown in the Figure 7.

In assessing the performance of attribute editing, we consider two key dimensions: image quality and attribute editing accuracy. Due to the absence of ground truth, we employ two alternative metrics to quantitatively evaluate our proposed model. To gauge the accuracy of facial attribute editing in our approach, we utilize a pre-trained attribute classifier which was also employed in AttGAN [13] and has a mean accuracy of 94.5% per attribute on the CelebA dataset. Our evaluation criteria involve comparing the predicted attribute of a generated image with the desired attribute. If the classifier's prediction matches the desired attribute, we classify the generation as correct; otherwise, it is deemed incorrect. Table 3 provides attribute editing accuracy for 13 attributes. Regarding image quality,

Table 3: Attribute Generation Accuracy

Attribute	Accuracy
Bald	76.46
Bang	93.54
Black Hair	91.42
Blond Hair	78.82
Brown Hair	87.34
Bushy Eyebrows	91.27
Eyeglasses	96.71
Gender	93.89
Mouth Open	97.12
Mustache	76.19
No Beard	93.08
Pale Skin	97.26
Age	87.01
Average	89.23

Table 4 : Reconstruction quality of different model

Method	PSNR/SSIM
IcGAN	15.28/0.430
FadarNet	30.62/0.908
AttGAN	24.80/0.819
Our Model	31.70/0.950

we maintain the target attribute vector identical to the source attribute vector and compared the reconstructed image with input image to present the PSNR/SSIM results for image reconstruction in Table 4. Our model, benefiting from the FTUs and differential attribute vectors, outperforms AttGAN [13] and ICGAN [5] significantly in terms of reconstruction quality. Notably, ICGAN exhibits limited reconstruction capability due to its training procedure. FaderNet [14] achieves superior reconstruction results, primarily attributed to each FaderNet model being trained to handle a single attribute.

For qualitative analysis, we compared our model with AttGAN [13], as shown in Figure 8. The results of AttGAN were produced utilizing the publicly available model. It is evident from these results that the AttGAN method still has limitations in manipulating sophisticated attributes such as baldness, hair, mustache, and age. The comparison reveals that our model exhibits superior reconstruction and attribute manipulation capabilities compared to AttGAN. This superiority can be attributed to the difference attribute vector, which helps to refine the features in FTU before transmitting them from the encoder layer to the decoder layer, whereas AttGAN employs a direct skip connection for feature reusability and full target attribute vector instead of difference attribute vector.

4. Conclusions

This Study demonstrates the efficacy of the use of differential attribute vectors as conditional information rather than full target attribute vectors for facial attribute editing. It is shown that the model generates visually realistic facial attribute edited images, preserving other attributes except for target attribute/s. The use of differential attribute vectors as conditional information leads to significant improvement in the model's accuracy because it only focuses on the attributes that need to be changed. This model outperforms the other state-of-the-art models regarding generated image quality and attribute generation accuracy.

Acknowledgment

This work was supported by Department of Electronics and computer engineering, TU, IOE, pashchimanchal campus, pokhara, Nepal.

References

[1] Sarker, I.H. Deep Learning: a comprehensive overview on techniques, taxonomy, applications and research directions, *SN Computer Science*, 2(6) (2021) 420.

[2] Theis, L., Oord, A. V. D. and Bethge, M. A note on the evaluation of generative models, *arXiv preprint arXiv:1511.01844,* (2015).

[3] Larsen, A. B. L., Sønderby, S. K., Larochelle, H. and Winther, O. Autoencoding beyond pixels using a learned similarity matrix, *International Conference on Machine Learning*, (2016) 1558-1566.

[4] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. and Bengio, Y. Generative adversarial nets, *Advances in neural information processing systems,* (2014) 27.

[5] Perarnau, G., Van De Weijer, J., Raducanu, B. and Álvarez, J. M. Invertible conditional gans for image editing, *arXiv preprint arXiv:1611.06355,* (2016).

[6] Mehdi, M. and Simon, O. Conditional generative adversarial nets, *arXiv preprint arXiv:1411.1784,* (2014) .

[7] Li, M., Zuo, W. and Zhang, D. Deep identity-aware transfer of facial attributes, *arXiv preprint arXiv:1610.05586,* (2016).

[8] Shen, W. and Liu, R. Learning Residual Images for Face Attribute Manipulation, *IEEE Conference on Computer Vision and Pattern Recognition,* (2017) 4030–4038.

[9] Zhou, S., Xiao, T., Yang, Y., Feng, D., He, Q. and He, W. Genegan: Learning object transfiguration and attribute subspace from unpaired data, *arXiv preprint arXiv:1705.04932,* (2017).

[10] Yin, W., Fu, Y., Ma, Y., Jiang, Y., Xiang, T. and Xue, X. Learning to Generate and Edit Hairstyles, *In Proceedings of the 25th ACM International Conference on Multimedia*, (2017) 1627-1635.

[11] Xiao, T., Hong, J. and Ma, J. Dna-gan: Learning disentangled representations from multi-attribute images, *arXiv preprint arXiv:1711.05415,* (2017).

[12] Kim, T., Kim, B., Cha, M. and Kim, J. Unsupervised visual attribute transfer with reconfigurable generative adversarial networks, *arXiv preprint arXiv:1707.09798,* (2017).

[13] He, Z., Zuo, W., Kan, M., Shan, S. and Chen, X. ATTGAN: Facial attribute editing by only changing what you want, *IEEE Transactions on Image Processing*, 28(11) (2019)5464–5478

[14] Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L. and Ranzato, M. A. Fader networks: Manipulating images by sliding attributes, *Advances in neural information processing systems,* (2017) 30.

[15] Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S. and Choo, J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation, *In Proceedings of the IEEE conference on computer vision and pattern recognition,* (2018) 8789-8797.

[16] Ronneberger, O., Fischer, P. and Brox, T. U-net: Convolutional networks for biomedical image segmentation, In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18,* Springer International Publishing. (2015) 234-241.

[17] Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K. and Rueckert, D. Attention u-net: Learning where to look for the pancreas, *arXiv preprint arXiv:1804.03999,* (2018).

[18] Arjovsky, M., Chintala, S. and Bottou, L. Wasserstein generative adversarial networks, In *International conference on machine learning,* (2017) 214-223.

[19] Brownlee, J. *Generative Adversarial Networks with Python: Deep Learning Generative Models for Image Synthesis and Image Translation*, Machine Learning Mastery (2019).

[20] Chen, D., Wei, K., Jiaqi, Z. and Shuangyan, D. InjectionGAN:Unified Generative Adversarial Networks for Arbitrary Image attribute editing, *IEEE Access,* (2020) 117726-117735.

[21] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980,* (2014).

[22] Liu, Z., Luo, P., Wang, X. and Tang, X. Large-scale celebfaces attributes (celeba) dataset.(2014) See https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html (Retrieved August 15, 2018).

[23] Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift, In *International conference on machine learning,* (2015) 448-456.