Disentanglement of Human Facial Features and the Inspection of Feature Extractor's Quality Based on TL-GAN

Disentanglement of Human Facial Features and the Inspection of Feature Extractor's Quality Based on TL-GAN

Yutong Gao[1,a,†], Shiwei He[2,b,†], Guanghan Wang[3,c,†] and Peiyu Xu[4,d,†]

[1]Information and Computer Science, Beijing University of Civil Engineering and Architecture, Beijing, 102612, China

[2]Safety Engineering, China University of Mining & Technology-Beijing, Beijing, 100089, China

[3]Computer Science and Technology, Tsinghua University, Beijing, 100084, China

[4]Telecommunications Engineering and Management, Beijing University of Posts and Telecommunications, Hebei, 065000, China

[*]Corresponding authors' e-mail: 1810130213@student.cumtb.edu.cn

[†]These authors contributed equally.

Abstract. Recent research on the Generated Adversarial Network models has made great progress in the task of generating images on human demand. Among them, Transparent Latent-space GAN tries to solve the problem by analyzing the latent space and finding the feature axes inside it. This model views the problem from a novel perspective, and it is easy to be carried out. However, more thorough and detailed experiments have shown that the original method on axes disentanglement of the model has certain drawbacks, which may lead to ethical problems, meaning that the model may pick up some pre-existing prejudice of racial or gender discrimination. This paper puts forward a new training mode aiming at reducing the bias existing in the generated images and simultaneously managing the disentanglement. Also, a more thorough examination of the model was conducted and it is found that the architecture of the model may be generalized to some universal functions. Among them, the quality evaluation of the feature extractor is the most practical and useful one. This finding, in turn, may be helpful for distinguishing a more accurate, effective, and robust feature extractor, thus improving the performance of the TL-GAN model in the first place.

1. Introduction

Generative Adversarial Networks (GAN) have performed well among variable computer vision tasks. For instance, Cycle-GAN has a marvelous and graceful performance in style transfer. Star-GAN established connections between multiple fields. Even there is a Multi-Content GAN (MC-GAN) that is able to transfer the style of the font. However, considering one style of the image a certain "feature", it would be a natural question that in what way can the generator generate images with certain features required, or identically, how can humans control the features of the images we generate.

By the research and work on seeking a better way to disentangle the correlated feature vectors as well as the idea that TL-GAN can be utilized as the quality inspector of the feature extractor, the strengthened result of the generator in GAN can be taken to another level. The improved model is not only capable of generating impressively photorealistic high-quality photos of faces, but also offers

control over the style of the generated image at different levels of detail through varying the style vectors and noise, and thus, strengthens the control over the synthesized image.

With the call of the significance of contributing to the development of the GAN, a vast number of extraordinary studies on this topic have been carried out by many researchers in recent years.

Goodfellow and others proposed the revolutionary idea–GAN, which makes two neural networks compete and collaborate. However, the training of GAN is unstable and the generation process is not controllable. In addition, it has no interpretability [1]. The discriminator and generator of Deep Convolutional Generative Adversarial Network (DCGAN) use the convolutional neural network (CNN) to replace the multi-layer perceptron in GAN to make the whole network differentiable. The pooling layer in CNN is removed, and the full connection layer is replaced by the global pooling layer to reduce the amount of computation [2]. Mirza and others proposed the Conditional Generative Adversarial Network (CGAN, which is a generative antagonism model with conditional constraints [3]. This conditional variable introduced in the modelling of generator and discriminator can be viewed as an improvement of the unsupervised GAN. Unlike GAN and CGAN, Cycle GAN can achieve migration between the source and target domains without establishing a one-to-one mapping between the training data [4]. Coupled Generative Adversarial Networks (CoGAN) trains a "couple" of GANs rather than a single one. Since it shares some of the weights, a CoGAN would have fewer parameters than two individual GANs [5]. The progressive growing Generative Adversarial Networks (PGGAN) is a technique that helps to stabilize GAN training by gradually increasing the resolution of the generated images [6]. Wasserstein Generative Adversarial Networks (WGAN) is proposed using Wasserstein distance as an optimization method to train GAN, which theoretically solves the problem of unstable training [7]. Self-attention Generative Adversarial Network (SAGAN) introduces the self-attention mechanism into convolution, showing a better balance among the ability to simulate remote dependencies, computational efficiency, and statistical efficiency [8]. BigGAN introduces a variety of techniques to combat the instability of training GANs on huge batch sizes across many machines [9]. StyleGAN is developed from Nvidia research that is mostly orthogonal to the more traditional GAN research, which focuses on loss functions, stabilization, architectures, etc [10].

This paper focuses on the model applying GAN, Neural Style, and its problems. Style GAN, which has been put forward recently, can implement control over the GAN-generated images. Intuitively, when using PGGAN to generate images, the layers handling low-resolution images are more related to these general and vague features，while the layers handling high-resolution images tend to concern relatively more precise features. TL-GAN tries to solve the problem from another perspective. By studying the latent space of the input vector (512 dimensions in the case of PGGAN), TL-GAN tries to figure out the axes for each feature. In addition, by adding the axes to the original vector, control can be implemented over a certain feature. The idea is innovative and commendable, but there are still some issues to solve in further research, among which the most severe one is the chaotic entanglement among axes, leading to some concerns in terms of both ethics and efficiency.

By analyzing the experiment data, a high-level correlation among features' axes is noticed, which means that the corresponding features are in great entanglement. Although Guan's straightforward linear algebra approach is intuitive and easy to be carried out, this method is not truly effective in solving the entanglement problem. Thus, a new method is put forward to implement the disentanglement. Besides, the correlation between feature axes is noticed to potentially serve as a way to indicate the performance of the feature extractor. Additionally, it is found that through implementing the Generalized Linear Model, the relation between the generated feature axes may partially explain what happens inside the feature extractor.

Overall, our contributions are as follows:

- We unveil the problem that biases exist in many training datasets which causes the issue of feature entanglement beforehand.
- We thoroughly inspect TL-GAN and make some refinement. We point out the drawback of TL-GAN's disentanglement method and put forward a new way to achieve the goal.

- We demonstrate how to evaluate the performance of the feature extractor by implementing the Generalized Linear Model and analyzing the correlation between generated axes.

2. Method

2.1. TL-GAN

Shaobo Guan's TL-GAN is a GAN model capable of generating images with certain features given the corresponding index number, providing a new method to control the generation process of unsupervised generative models. TL-GAN uses a trained GAN generator initially, and then substantialize the meaningful feature axes to make the process of generating an image transparent, which makes the control over the process of image synthesis and editing highly practical.

2.2. Make the potential space behind the image transparent

NVIDIA's PGGAN is a deep learning model highly efficient in generating high-resolution images. Based on the intuition of growing the capacity of the networks progressively, PGGAN starts with a 4×4 pixels spatial resolution in both the Discriminator and the Generator, iteratively adds a double-sized convolutional layer in both networks respectively. As shown in Fig.1, Guan finds that the latent space of PGGAN is densely populated and smoothly connected. The former means that most points in the latent space are able to generate a certain image and the latter means that when transiting between two vectors, the corresponding images they generated tend to change smoothly.

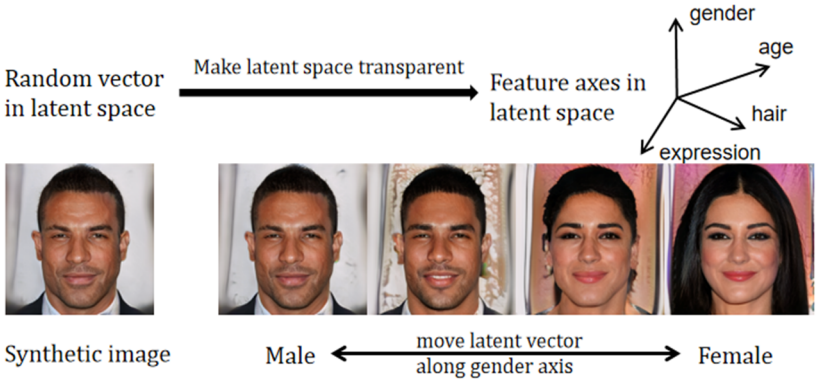

Figure 1. Transparency of latent space to control the generation process

2.3. Find feature axis

Guan builds a separate feature extractor model y=F(x) (a classifier for the dispersed label or a regressor for the continuous label), which is trained by using the existing annotated image data set (x_real, y_real) to train, and concatenates it to the generator, so that we can use the trained feature extractor network to predict the feature label y_pred of the composite image x_gen, so as to establish the relationship between z and y demonstrated in Fig.2, that is, x_gen=G(z) and y_pred=F(x_gen). Thus, forming a regressor y=A(z) to find out the unit vectors in the latent space corresponding to the image features. By adding multiples of these unit vectors to the original vector, TL-GAN could generate images with continuously changing features. The Generalized Linear Model (GLM) is used to perform the regression task between latent vectors and features. The regression slope is the feature axis.

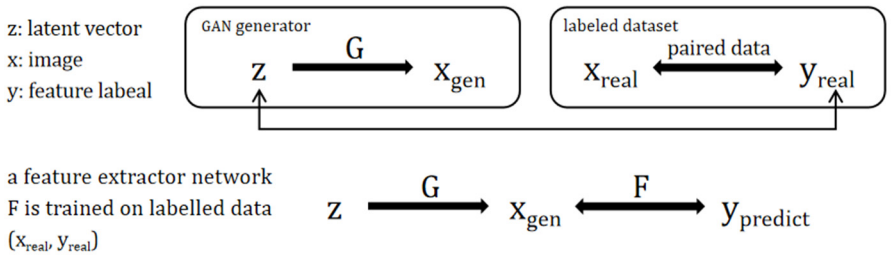

Figure 2. A method of connecting the latent vector z and the feature label y

2.4. Move latent vectors along the feature axis

A random vector z_0 is generated in the latent space of GAN and then transmitted to the generator network x_0=G (z_0) to generate the composite image x_0. Next, move the potential vector along the feature axis u (the unit vector in the potential space, corresponding to the gender of the face) by a distance λ to a new position x_1 = x_0 + λu, and a new image x1 = G(z1) is generated. The following figure shows the results of moving potential space vectors along multiple sample feature axes (gender, age, etc.).

2.5. Disentangle axes

Given the fact that some features may be naturally correlated (beard and gender for instance), Guan applies the straightforward linear algebra method to orthogonalize the unit vectors, thus managing to disentangle the correlated feature axes.

3. Analysis and solution

To illustrate the feature entanglement issue, we experiment with the GUI of the TL-GAN model by increasing the feature value of "Blond", consequently the feature of "Female" also becomes distinct as demonstrated in Fig.3. Thereupon, we can draw a conclusion that some of the features such as hair color and gender are badly calibrated and biased which are not accurate enough to alter the image towards the specific direction even after being processed with the linear algebra method put forward by Guan—Orthogonalization.

To solve the issue of feature entanglement, there're two approaches backed up by the experiment undermentioned.

Firstly, it is discovered that the feature entanglement issue not only occurs because the feature vectors aren't extracted accurately but also relates to the uneven feature distribution in the dataset. Nonetheless, the problem can be fixed by using datasets consisting of balanced feature distribution to training the model as what we do in the experiment.

Figure 3. Feature entanglement demonstration

As CelebA is chosen as the training dataset, the samples are divided into plenty of batches according to some prominent features of the images. In achieving the goal of features disentanglement regarding the biased dataset, the only tool needed is the classifier telling to what extent each batch is distributed unevenly.

Secondly, the experiment to test the quality of different feature extractors is conducted, four different feature extractors are prepared to be assigned to a TL-GAN for comparison of the generated images performance.

3.1. Disentanglement and its role in bias reduction

With the swift development of deep learning, artificial intelligence has been increasingly trusted to advise on some significant decision and selection work. One may not get a job interview, for instance, if he or she is judged incompetent by the Artificial Intelligence. However, due to the commonly existed bias in human society and the vast scale of training data, AI may pick up those prejudices without being known. We have found that such a problem exists in the TL-GAN model. For example, when increasing the width of one's chin, the person tends to get balder and the skin becomes darker (illustrated in Fig.4), which indicates that white people's faces are more relative with the elements that make a person attractive. Feature disentanglement may be a solution to this and the ideal state is that when tuning the width of one's chin, the color of the skin is unchanged, for the feature axes of "chin" and "skin color" are perpendicular to each other. Nevertheless, our experiment shows that even after disentanglement the bias still exists, which in turn indicates that Shaobo Guan's straightforward linear algebra method is not effective enough.

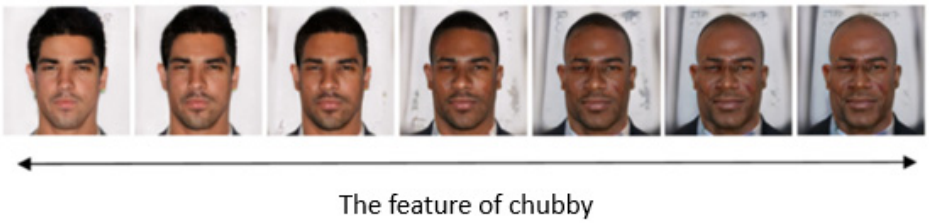

Figure 4. The baldness of the image changes as the width of the chin change

The major reason behind this phenomenon, as we assumed, is the wide-spread unbalance in the training datasets. For instance, when discussing the unpleasant correlation between skin color and the width of one's chin, it is clear that there are a great many pictures of black people with a wide chin (marked with the image with property A) while the pictures in which a black person has a narrow chin (marked with images with property B) are rare. Given the fact, provided that images with property B are injected into the mini-batch to an extent that the number of images with property B is roughly equal to that of images with property A, the unhealthy correlation will be drastically reduced.

Hence, we put forward a new training mode, Data Balance among Features, or DBF in short. Firstly, pictures are classified by some significant features, race, and gender for instance, and each group forms a batch. Within each batch, unbalance is detected feature by feature iteratively with a pre-trained binary classifier. If the ratio between the size of two categories surpasses the threshold, the data group with the smaller size will be augmented. Augmentation methods such as small-angle rotation, picture reversal, and some more sophisticated ones can be utilized. Also, other methods such as straightforward orthogonalization or vector debiasing may be combined with DBF to strengthen the disentanglement effect [11].

3.2. Using TL-GAN to evaluate the quality of feature extractors

After the axes of different human face features generated by the trained feature extractor are applied to the random noises, the random noises become oriented. Axes of several requested features are formed, thus the input becomes transparent, like what the model creator calls "Transparent Latent Space".

In the seeking of resolving the problem of features entanglement, from a different perspective, it occurred to us that rather than using the algebra approach of orthogonalizing tangled feature vectors to disentangle, why don't we make the vectors uncorrelated in the first place? Upon the notice, we discovered that the quality of the feature extractor highly affects the image accuracy. Based on that opinion, we can replace the current feature extractor with a better one to improve the network performance. Moreover, we can yet make the TL-GAN a quality inspector of feature extractors as referred to in Fig.5.

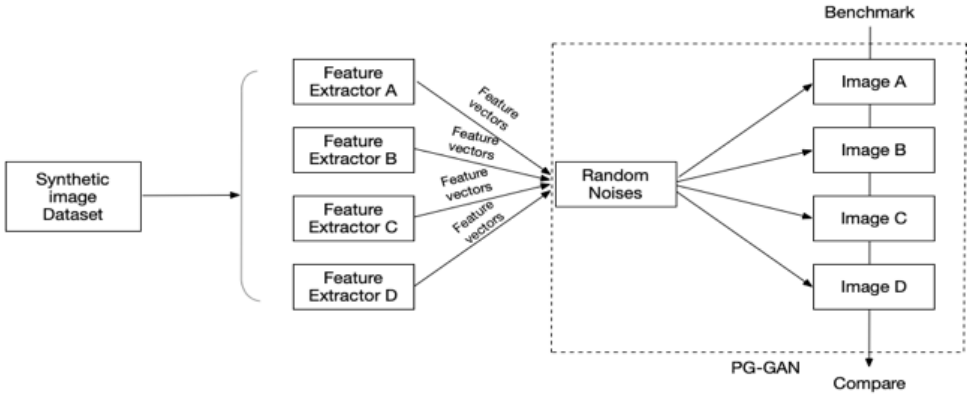

Figure 5. Using TL-GAN as a quality inspector

On that account, we concluded that, on the foundation of a well-trained PGGAN, the quality and effectiveness of the feature extractor can be straightly reflected by the running results of the TL-GAN. The more perfect the feature extractor gets close to, the more accurate and impeccable the generated images would be along the requested features axis. Conversely, the less descriptive and more tangled the images with requested features would become.

With that being said, we experimented with the model using different feature extractors to compare the results.

3.2.1. Experiment group. To evaluate the condition of different feature extractors, a reference indicator is required. Thereupon, we set the feature “Bald” to a specified value, and through some other methods like manually adjusting the feature value, the reference image is managed to remain the original value along other feature axes. Thus, the image becomes the benchmark. In the experiment group, we choose feature extractor A as our subject. In GUI, we increased the feature value of "Bald" to a certain point, then modified some other features like "Male" accordingly to allow the image to move accurately along the “Bald” axis without being tangled with other features.

3.2.2. Control group. In the control group, we replaced the feature extractor A with another three—Feature extractor B, C, and D. Likewise, we adjusted the feature value of "Bald" by increasing it at the matching level with the experiment group. To our notice, even though processed with the same PGGAN and the “Bald” feature values are assigned with the same number, the images generated individually by the feature extractor B, C, and D appear at different levels of feminine-looking as results displayed in Fig. 6.

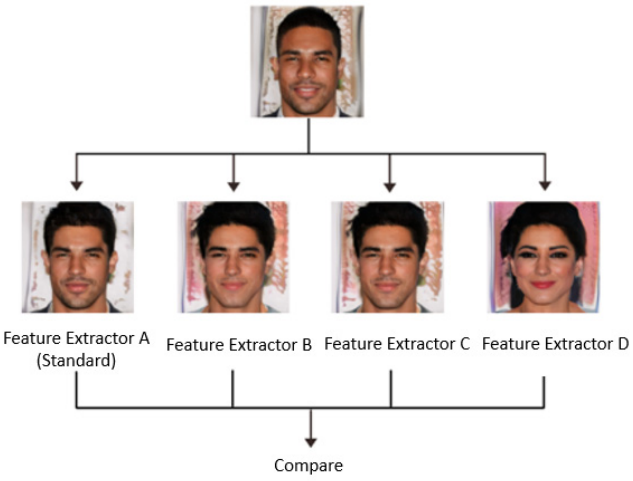

Figure 6. Images for comparison

In conclusion, by applying different feature extractors to the same TL-GAN to edit face attribution and comparing the corresponding synthetic images, we can make TL-GAN function as a method of measuring the feature extractor's quality. Namely, make TL-GAN the quality inspector for different feature extractors.

4. Conclusion

In this paper, a thorough examination of Shaobo Guan's TL-GAN model is conducted and the drawback of Guan's original straightforward linear algebra approach is pointed out in an effort to disentangle the feature axes. Also, it is found that the entanglement among axes may result in the model picking up some pre-existing human society prejudice. Hence, a new training mode is put forward to primarily try to improve the representativeness and reduce the bias in the training set. Not only the architecture of TL-GAN is highly inspiring, but it also serves as a feature extractor quality inspector. In turn, experiments have shown that such a finding helps to improve the model performance. We hope that our work is helpful for those who are trying to refine the generator of the GAN model and it may provide some intuition for researchers who are working to reduce the bias in AI models.

References

[1] Goodfellow, Ian J., et al. (2014). "Generative Adversarial Networks." In: Neural Information Processing Systems 3: 2672-2680.
[2] Radford, A., Metz, L., & Chintala, S.. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In: ICLR. San Juan.
[3] Mirza, M., & Osindero, S.. (2014). Conditional generative adversarial nets. In: Computer Science. pp. 2672-2680.
[4] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. Venice. pp. 2223-2232.
[5] Liu, M. Y., & Tuzel, O.. (2016). Coupled generative adversarial networks. In: NIPS. Barcelona. pp. 469-477.
[6] Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. In: ICLR. Vancouver.
[7] Arjovsky, M., Chintala, S. & Bottou, L.. (2017). Wasserstein Generative Adversarial Networks. Proceedings of the 34th International Conference on Machine Learning. In: PMLR 70:214-223.

[8] Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. In: International conference on machine learning. pp. 7354-7363.
[9] Brock, A., Donahue, J., & Simonyan, K.. (2018). Large Scale GAN Training for High Fidelity Natural Image Synthesis. In: ICLR. New Orleans. pp. 4356-4364.
[10] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4401-4410.
[11] Bolukbasi, T., Chang, K. W., Zou, J., Saligrama, V., & Kalai, A.. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In: NIPS. Barcelona. pp. 4349-4357.