Semantic Face Editing with GAN Latent Code Optimization

Martin Pernuš, Vitomir Štruc, Simon Dobrišek

Faculty of Electrical Engineering, University of Ljubljana
E-pošta: {martin.pernus, simon.dobrisek}@fe.uni-lj.si

Abstract

Recent advances in generative modeling of images have reached new heights in the task of image generation. The generative model that produces the most photorealistic and high resolution images is Generative Adversarial Network (GAN). In our paper we propose a multi-task latent vector optimization procedure that combines local GAN inversion with face attribute constraints for the task of semantic face editing. The experiments show that the method produces visually pleasing and semantically corresponding face images.

1 Introduction

With the advancements in generative modelling with Generative Adversarial Networks (GANs) [5], image generation has achieved unprecedented image photorealism. The latent space of GAN models provides a compressed representation of high-resolution image, which can be useful for semantic image manipulation. By manipulating the latent codes of GAN generated images, several useful image editing applications have been proposed [1, 2, 19]. The advancements in this field could have an important impact toward automatic image editing tasks. Such algorithms could enable image manipulation by simply specifying the desirable visual attributes. This could have an important impact in art, media and entertainment industry.

The process of obtaining a latent code of an image and performing various operations has been popularized by the variational autoencoder [12]. Research showed that combining latent codes produced a learned image manifold that did not just correspond to pixelwise averaging, but had a sense of the intrinsic data distribution. Since the GAN architecture lacks a direct way to obtain an image's latent code, GAN inversion optimization method [1, 2] is usually applied.

Generative modeling and image editing tasks have been largely focused on face modeling and editing as generative modeling relies on vast quantities of training examples. For example, state-of-the-art StyleGAN model [9] is trained on 70,000 high resolution face images. Several studies used the pretrained StyleGAN model to edit face images. Our work is inspired by the work of [19], where the latent code vector space of GANs was analyzed and linear movements in vector space were proposed to edit face images. Instead of analyzing latent space, we directly perform gradient-based optimization on latent code with spatial and semantic constraints.

Although studies have analyzed latent code manipulation for various semantic image operations, disentanglement of semantic variations of latent codes remains challenging. In our work, we propose to use a local GAN inversion technique to preserve the person's identity while modifying the selected face attributes. We test the proposed method and visually analyze the results in comparison with a competing method.

2 Related Work

Generative Adversarial Networks (GANs) are one of the most used generative models for image modelling task. Since the original proposal in [5], the majority of advances stem from better architectures and loss functions. The architecture design was improved in [18], which proposed convolutional GAN design. Karras et al. [10] first managed to produce megapixel images using progressive learning of GANs. The model was further improved in StyleGAN [8, 9], where the architecture is inspired by style transfer architecture, resulting in state-of-the-art unconditional image generation. Loss function beyond the one proposed in the vanilla GAN model were explored in [6, 14, 16, 17].

Due to resource intensive training of GAN models, recent research focuses on the analysis of pretrained GAN models. Pretrained parametric space of GAN weights was manually tuned in [3] to achieve localized deletion and addition of objects in the output image. In [7] linear and non-linear walks in latent space were learned that achieved some basic image manipulation, such as brightness and zoom change. In [19], linear subspaces of latent facial semantics were identified to edit face images.

These methods are based on GAN generated images. GAN models are inherently constrained by the probability distribution of the data they are trained on, which limits the usability of latent codes for image editing. Abdal et al. [1, 2] projected face images using GAN inversion technique in an extended latent space before editing face images in various ways. In our work, we also use a version of GAN inversion with an extended latent space, but we also consider various spatial and semantic constraints.

3 Methods

Our method is based on StyleGAN model [9]. StyleGAN is the current state-of-the-art GAN method for unconditional image generation. We use the pretrained StyleGAN model trained on Flickr-Faces-HQ dataset [8].

StyleGAN's generator is defined by two main components: initial latent code non-linear mapping and the generator G that generates the final image. Non-linear latent code mapping is defined as $f : \mathcal{Z} \rightarrow \mathcal{W}$ that maps the initial Gaussian sampled latent code z to code w of the same 512-d dimensionality. The purpose of mapping f is the disentanglement of factors of variation that could be present in the Z vector space due to its Gaussian distribution. The generator G maps latent space $\mathcal{W}$ to images. The w codes are mapped by 18 learned affine transforms as an input to the convolutional layers of the generator. Each convolutional layer also receives a stochastic component in the form of spatial noise $n \in \mathcal{N}_S$ that slightly affects the output face image. See [9] for more details about the model.

The goal of GAN inversion is retrieving the latent code w and optionally n that best matches the image I given pretrained generator G. As shown in [2], a wide variety of images, including non-face images, can be embedded in the extended latent space W^+. The extended latent space W^+ consists of a concatenation of 18 different 512-dimensional w vectors. We also optimize the noise component of StyleGAN n.

To achieve the presence of selected face attribute on the final image, we also introduce a pretrained classifier C that predicts the presence of the selected facial attribute. The predicted probability of the selected attribute is denoted as $\hat{y}(w, n) = C(G(w, n))$.

Starting from a suitable initialization of w and n, we search for their optimized versions w^* and n^*. We only optimize a subset of noise n, the portion that affects the face portion of the image. Our loss function is defined as

$$\begin{aligned}\mathcal{L}(w, n) =& \lambda_{\text{mse}} ||M \odot (G(w, n) - I)||_2^2 + \\ & \lambda_{\text{ce}}(-y \log \hat{y}(w, n) - (1 - y) \log(1 - \hat{y}(w, n)))\end{aligned} \tag{1}$$

where M is the spatial mask that defines the region where mean squared error part of the loss is defined, I is the target image, λ_{mse} is the weighting constant for mean squared error loss and λ_{ce} is the weighting constant for the binary cross entropy loss.

A special consideration must be taken with regard to the spatial mask M. It must be large enough to preserve enough face in the image to retain the face identity, while also allowing the rest of the image to change enough to satisfy the selected face attribute constraint. We define M with face segmentation model that is based on the DeepLabv3 segmentation model [4]. We trained the model to spatially predict the face components of interest on the image I based on the selected facial attribute. We then binarize the prediction based on predefined threshold and blur it with Gaussian filter to allow smooth changes to the face image.

4 Experiments

4.1 Implementation

The classifier C is based on the state-of-the-art multi-task neural architecture [20] and is trained on CelebA dataset [15] for 23 epochs. The starting learning rate is 0.05 and it decays to one tenth its current value every 40,000 steps. The CelebA dataset contains 200,000 celebrity images with 40 annotated facial attributes per image as well as their identity information.

The face segmentation model is set as DeepLabv3 model [4] that is trained on CelebAMaskHQ dataset [13]. This dataset contains 30,000 images with the size of 512×512 and 19 facial components and accessories such as skin, nose, eyes, eyebrows, ears, mouth, lip, hair, hat, eyeglass, earring, necklace, neck, and cloth. Each image is annotated with a segmentation mask of facial components. After grouping several semantically similar attributes, we ended up with 'background', 'mouth', 'eyebrows', 'eyes', 'earrings', 'hair', 'nose' and 'skin' facial components that we use to train the face segmentation model. The model is trained for 5 epochs with a learning rate of $3 \cdot 10^{-4}$.

The latent code w is initialized with the latent code mean, calculated by passing 10,000 Gaussian sampled z vectors and calculating the mean of the StyleGAN's feedforward network. The optimization of w^* and n^* is run for 2000 iterations. The learning rate is first linearly increased from 0 to 10^{-2}, then decayed back to 0 using cosine schedule during the last 500 iterations. The threshold for binarizing the image is set to 0.9 and the size and the standard deviation of Gaussian filter are set to 51 and 31, respectively. λ_{mse} is set to 100 and λ_{ce} is set to 10.

The optimization algorithm of choice for all the models is the Adam optimization method [11].

4.2 Evaluation

For comparison with the existing face editing techniques, we implemented the Interface method [19]. The main idea behind the Interface method is finding a hyperplane that optimally separates latent space based on support vector machine and moving the latent code in the selected direction. It operates in W space, which guarantees that latent code will generate a face image. However, W space is often lacking when performing GAN inversion. Thus, the reconstructions do not match the original image as well as they would with W^+ space.

To find the individual facial attribute hyperplane we followed the Interface method procedure. First we generated 500,000 StyleGAN images. We picked the 10,000 most positive and 10,000 most negative images per facial attribute according to our classifier C. The hyperplane was then identified using linear support vector machine on the latent codes of these images. For visualization we move the latent codes for 0.5 the norm of the latent vector as opposed to 3 times the norm in the original proposal, since in our experiments the face images failed to preserve any identity information in the latter case.

In Figure 1 we show the results of the compared Interface method and our proposed method for several facial

attributes. The results visually indicate that our proposed method retains the identity information better than the Interface method.

The advantages of our method in comparison with InterFace method can be summarized as better visual results and no pretraining requirement. However, once the InterFace learns the optimal hyperplane, its method allows immediate calculation of new latent vectors (and new face images), while our method requires optimization of latent code. Our optimization procedure takes approximately 10 minutes.

5 Conclusion and future work

In this paper, a method for changing facial attributes while preserving identity is proposed. The method is based on local GAN inversion technique with facial attribute constraints. Experimental results visually suggest that real face images can be modified according to selected attributes while preserving their identity.

Additional improvements could be made using disentanglement techniques, which could modify latent space in a way that changing a single facial attribute wouldn't affect other attributes, while allowing the pose of the person to change. That is the topic of our further research.

Acknowledgments

This work was supported in parts by the ARRS Research Programmes P2-0250 (B) Metrology and Biometric Systems, the ARRS young researcher program and the ARRS research project J2-2501 - Deep generative models for beauty and fashion (DeepBeauty).

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. "Image2StyleGAN++: How to Edit the Embedded Images?" In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8296–8305.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. "Image2stylegan: How to embed images into the stylegan latent space?" In: *Proceedings of the IEEE international conference on computer vision*. 2019, pp. 4432–4441.

[3] David Bau et al. "GAN Dissection: Visualizing and Understanding Generative Adversarial Networks". In: *International Conference on Learning Representations*. 2018.

[4] Liang-Chieh Chen et al. "Rethinking atrous convolution for semantic image segmentation". In: *arXiv preprint arXiv:1706.05587* (2017).

[5] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[6] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *Advances in neural information processing systems*. 2017, pp. 5767–5777.

[7] Ali Jahanian, Lucy Chai, and Phillip Isola. "On the" steerability" of generative adversarial networks". In: *International Conference on Learning Representations*. 2019.

[8] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410.

[9] Tero Karras et al. "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8110–8119.

[10] Tero Karras et al. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *International Conference on Learning Representations*. 2018.

[11] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[12] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[13] Cheng-Han Lee et al. "Maskgan: Towards diverse and interactive facial image manipulation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 5549–5558.

[14] Jae Hyun Lim and Jong Chul Ye. "Geometric gan". In: *arXiv preprint arXiv:1705.02894* (2017).

[15] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[16] Xudong Mao et al. "Least squares generative adversarial networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2794–2802.

[17] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. "f-gan: Training generative neural samplers using variational divergence minimization". In: *Advances in neural information processing systems*. 2016, pp. 271–279.

[18] A. Radford, L. Metz, and S. Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[19] Yujun Shen et al. "Interpreting the latent space of gans for semantic face editing". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9243–9252.

[20] Simon Vandenhende et al. "Branched multi-task networks: deciding what layers to share". In: *arXiv preprint arXiv:1904.02920* (2019).

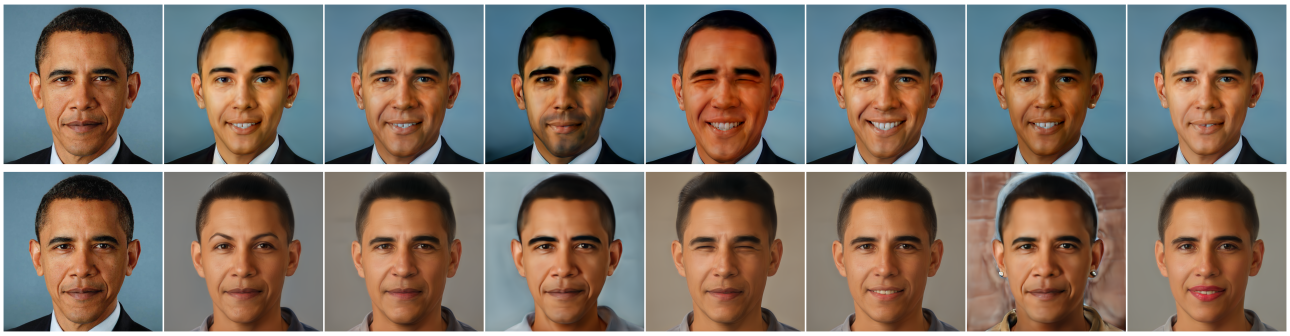

Figure 1: Comparison of InterFace [19] performance (first row) to our method (second row) on several face editing tasks for an example face image. The selected facial attributes from left to right are defined as follows: original image, arched eyebrows, big nose, bushy eyebrows, narrow eyes, smiling, wearing earring and wearing lipstick.

Figure 2: Comparison of InterFace [19] performance (first row) to our method (second row) on several face editing tasks for an example face image. The selected facial attributes from left to right are defined as follows: original image, blond hair, brown hair, gray hair, straight hair, wavy hair.