Inverse mapping of face GANs

1 How to replicate our results

1.1 Abstract

Generative adversarial networks (GANs) synthesize realistic images from a random latent vector. While many studies have explored various training configurations and architectures for GANs, the problem of inverting a generative model to extract latent vectors of given input images has been inadequately investigated. Although there is exactly one generated image per given random vector, the mapping from an image to its recovered latent vector can have more than one solution. We train a ResNet architecture to recover a latent vector for a given face that can be used to generate a face nearly identical to the target. We use a perceptual loss to embed face details in the recovered latent vector while maintaining visual quality using a pixel loss. The vast majority of studies on latent vector recovery perform well only on generated images, we argue that our method can be used to determine a mapping between real human faces and latent-space vectors that contain most of the important face style details. In addition, our proposed method projects generated faces to their latent-space with high fidelity and speed. At last, we demonstrate the performance of our approach on both real and generated faces.

1.2 Inverse mapping of generated faces

In order to generate fake faces from random latent vectors, we use a progressive growing GAN trained on 128x128 CelebA faces [Karras et al.2017]. The trained model can be downloaded from tensorflow hub. Here is the link to download progan from tensorflow hub The model recieves 512 size random vectors sampled from normal distribution and generates fake faces. You can run below command in order to generate 50 fake faces that include faces used in the paper.

```
python generate_faces.py
```

In this work, we first try to map face images into latent-space using a ResNet18 architecture. The proposed framework is represented in Figure 1.

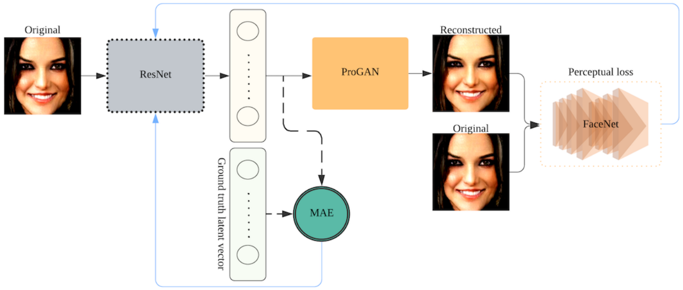

Figure 1: The proposed framework for mapping generated faces to latent-space vectors.

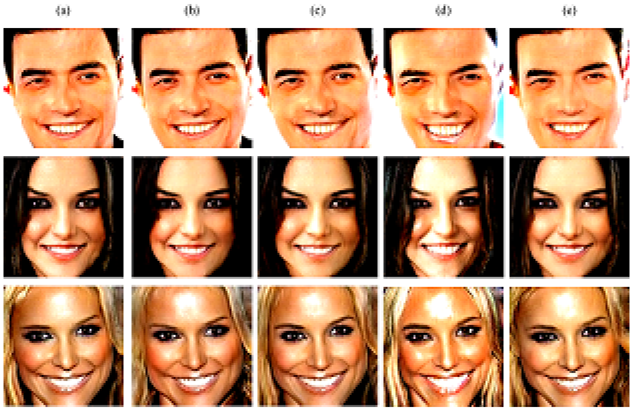

Figure 2: Original generated faces are presented in column (a). Column (b) is the generated faces of recovered latent vectors by (a) using gradient-descent method with 200 iterations. Column (c) applies stochastic clipping while updating gradient descent. Column (d) is our method, which utilizes our trained ResNet to map generated images to their corresponding latent vectors. Column (e) is our ResNet trained using both pixel and perceptual loss.

In order to train the ResNet18, we generated 100k faces using progan (you can use generate_faces.py to do the same) to use as train dataset. ResNet18 is trained based on a combination of z-loss (The loss between features extracted from last layer of ResNet and ground truth z vectors) and perceptual loss between original generated faces and the reconstructed peers. Perceptual loss is computed based on MSE loss between features of all concatenation layers of FaceNet [Schroff, Kalenichenko, and Philbin2015] for reconstructed and original faces. You must refer to the address of your 100k generated faces in PATH_TRAIN_generated. You should also save the corresponding z vectors for the training set in ”z_vectors/generated_face_100k.npy”

- You can download FaceNet pre-trained on MS-Celeb-1M from this link.
- ResNet18 architecture can be downloaded from this link.
- PATH_TRAIN_real is not required at this stage.
- Select a proper log and checkpoint directory.

In order to train ResNet, you can run below command:

```
python resnet.py
```

In order to recover latent vectors of generated faces, download our pretrained model from this link and save it in directory ”saved_model”.

- Next, run python generate_faces.py to generate 50 fake faces.
- Then run python resnet_test.py to save reconstructed faces in results folder.

Results of our pre-trained model are equal to column (e) of Figure 2. In order to obtain results for optimization-based alternatives [Lipton and Tripathi2017], you can run below command. You can comment stochastic clipping lines to replicate column (b).

```
python gradient_based_recovery.py
```

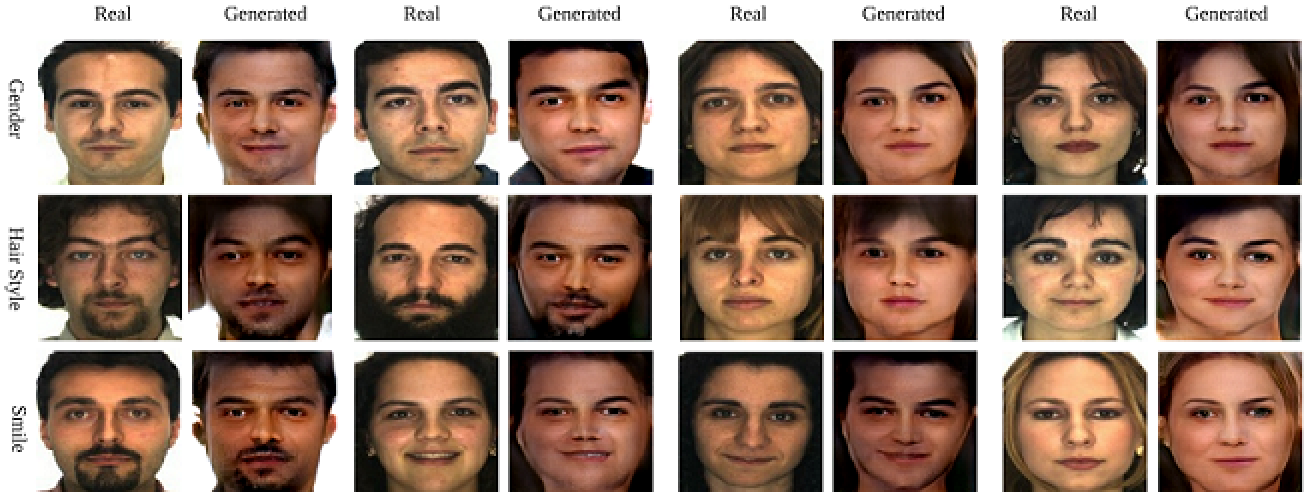

Figure 3: The results for mapping natural faces to latent-space vectors that contain same style and facial features. The faces in each row represent how recovered latent vectors understand the gender, hair style and emotions of the target image respectively.

1.3 Real faces

Real faces dataset: Select all VggFace2 [Cao et al.2018] identities that have more than 100 samples. Then select 100 samples for each identity randomly. Next, select 100 random identities, 10000 Vggface2 images will remain. Align and crop all faces using MTCNN [Zhang et al.2016]. PATH_TRAIN_real is the directory that contains these images. VggFace2 dataset can be downloaded from this link.

Now that you have your real faces dataset, you can train ResNet using python resnet.py. All you have to do is to uncomment real faces section in fit function. PATH_TRAIN_real must refer to the directory of vggfaces images. You can also use our pre-trained model from this link to replicate same results as Figure 3.

You must download AR dataset [Martinez1998] and align and crop them using MTCNN. AR dataset is used to evaluate the model in style transfer task and will replicate below figure. This dataset can be downloaded from this link.

Please make sure to change the model in resnet_test.py to the real faces model you just downloaded. Moreover, PATH_TEST should refer to AR faces now.

After downloading the model (the saved checkpoint) as well as AR dataset, the you can run python resnet_test.py to achieve below results.

```
Please contact me if any further assistance is required.
nbayat5@uwo.ca
```

References

[Cao et al.2018] Cao, Q.; Shen, L.; Xie, W.; Parkhi, O. M.; and Zisserman, A. 2018. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 67–74. IEEE.

[Karras et al.2017] Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

[Lipton and Tripathi2017] Lipton, Z. C., and Tripathi, S. 2017. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*.

[Martinez1998] Martinez, A. M. 1998. The ar face database. *CVC Technical Report24*.

[Schroff, Kalenichenko, and Philbin2015] Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

[Zhang et al.2016] Zhang, K.; Zhang, Z.; Li, Z.; and Qiao, Y. 2016. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters* 23(10):1499–1503.