Realistic Face Reconstruction from Deep Embeddings

Edward Vendrow
Department of Computer Science
Stanford University
evendrow@stanford.edu

Joshua Vendrow
Department of Computer Science
University of California, Los Angeles
jvendrow@math.ucla.edu

Abstract

Modern face recognition systems use deep convolution neural networks to extract latent embeddings from face images. Since basic arithmetic operations on embeddings are needed to make comparisons, generic encryption schemes cannot be used. This leaves facial embedding unprotected and susceptible to privacy attacks that reconstruction facial identity. We propose a search algorithm on the latent vector space of StyleGAN [7] to find a matching face. Our process yields latent vectors that generate face images that are high-resolution, realistic, and reconstruct relevant attributes of the original face. Further, we demonstrate that our process is capable of fooling FaceNet [11], a state-of-the-art face recognition system.

1 Introduction

Biometric authentication systems (i.e. face recognition) are extensively used for security. Such systems generate a template from a biometric data sample and compare it against a master template to provide authentication. These templates are often generated by incomprehensible black-box models, previously thought to be impossible to meaningfully deconstruct [5]. Nevertheless, recent works have succeeded in performing privacy attacks which extract soft biometric attributes or even full reconstructions from face embeddings [4, 14, 2, 3, 8].

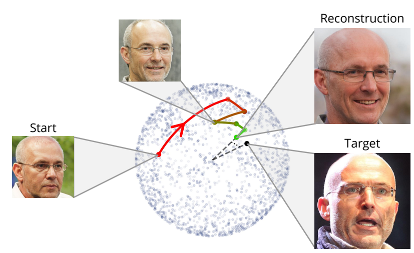

Figure 1: Reconstructing a target face with pre-generated initialization and simulated annealing on an example from the FFHQ data set. The path shows the progression of the search algorithm in the hyper-spherical embedding space as it finds a close match. For the purposes of visualization, we project the embeddings to a 3D spherical representation.

Reconstruction of face embeddings poses a major security risk from privacy attacks. Malicious attackers may access a database of embeddings and estimate a user's facial image or extract soft biometric attributes. The use of cryptographic methods to encrypt face embedding has not found much traction in the biometrics and pattern recognition community [2]. Generic encryption schemes such as one-way hashes are inherently incapable of supporting basic arithmetic operations in the encrypted domain, which is necessary for template matching [10]. Homomorphic encryption methods allow basic arithmetic operations over encrypted data, enabling encryption of face embeddings [2].

Previous works have demonstrated successful privacy attacks that fool face recognition systems with facial image reconstructions [4, 3]. However, the reconstructions are generally low-quality and would not convince a real human. Moreover, these methods would not work on homomorphically encrypted data. We propose a method which creates reconstruction face images that not only fool face recognition systems, but are also life-like and highly detailed. Our method does not require gradient information or white box access, which are unavailable for homomorphically encrypted data.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Figure 2: For 10 target images from the FFHQ data set, we display the reconstructed faces achieved by each of the four parameter settings: a) no pre-generation and greedy, b) pre-generation and greedy, c) no pre-generation and simulated annealing, d) pre-generation and simulated annealing.

1.1 Related work

The problem of recreating a face from its face embedding has drawn interest from the community.

Direct synthesis Zhmoginov et. al. [14] invert the FaceNet embedding of a face image using a guiding image to create a reconstruction capturing important identity features of the original face. Cole et al. [3] propose an autoencoder structure to map the features to a frontal, neutral-expression image of the subject. Yang et al. [13] train a second neural network for the model inversion task. These approaches require white-box access to the face recognition model to produce better quality results. Working in the black-box setting assumption, Mai et al. [9] propose a de-convolutional network framework to reconstruct face images without knowledge of the face recognition network.

Template Reconstruction with GANs Generative adversarial networks (GANs) have shown stunning results in generating convincing images [6]. StyleGAN [7], a style-based GAN, is able to generate high quality, artificial face images which are almost impossible to tell apart from real images. Recently Abdal et al. [1] demonstrated that it is possible to accurately embed arbitrary images onto the StyleGAN latent space. This suggests for any face embedding, it should be possible to find a StyleGAN latent vector whose corresponding face image has a nearly identical embedding. Li et al. [8] were the first to attempt this by iteratively improvement on a latent vector guess by adding random noise and greedily improving the guess based on the FaceNet embedding distance to the target. More recently, Duong et al. [4] propose a GAN-based system to reconstruct faces using metric learning methods. These methods reconstruct faces to fool some face recognition system, but the generated face images are generally low-quality and would not fool a human.

FaceNet FaceNet [11] is one of the most popular face recognition systems, using a deep CNN to map images onto a embedding space where squared L_2 distances directly correspond to face similarity. For two embeddings E_1, E_2, this face distance metric is defined as

$$D(E_1, E_2) = \|E_1 - E_2\|_2^2$$

with a classification threshold of 0.6, commonly set for implementations of FaceNet. Since output embeddings are normalized to magnitude 1, the embeddings are constrained to a d-dimensional hypersphere. Figure 1 shows a representation of the hyperspherical FaceNet embedding space as our reconstructed face image's identity iteratively approaches a target.

2 Methods

We propose the use of greedy random optimization within the latent space of StyleGAN to generated an image whose FaceNet embedding closely matches a target embedding, measured by the FaceNet distance metric. Starting with the zero vector, we repeatedly generate a new guess by adding small random noise and running the resulting vectors through the generator and FaceNet. We set a new guess to our current 'best' vector if it improves upon the previous 'best'. By decreasing the standard deviation of the input noise, we converge on a solution. We assess guesses via the FaceNet distance metric by comparing them to the target embedding.

We also aim to improve upon greedy random optimization by optimizing using simulated annealing [12] to find an even closer match. Simulated annealing will, with a probability depending on the difference between the loss of the current vector and new guess, accept a guess that is worse than the current vector in order to encourage exploration and 'hill climbing' over local minima. In Algorithm 1 we display the pseudocode for both of these optimization methods, where we denote StyleGAN as G and FaceNet as f.

Algorithm 1 Face Reconstruction

Paremeters: $\gamma \in [0, 1]$, standard deviation decay rate
Options: $\texttt{pregen} \in \{\texttt{T}, \texttt{F}\}$, $\texttt{anneal} \in \{\texttt{T}, \texttt{F}\}$
Initialization:
$$\mathbf{x}_{best} = \begin{cases} \vec{0}\,, & \text{if} \quad \texttt{pregen} = \texttt{F} \\ \texttt{Closest}(E)\,, & \text{if} \quad \texttt{pregen} = \texttt{T} \end{cases}$$
for $t \leftarrow 1, n$ **do**
$$T = \begin{cases} 1\,, & \text{if} \quad \texttt{anneal} = \texttt{F} \\ 1 - (t+1)/n\,, & \text{if} \quad \texttt{anneal} = \texttt{T} \end{cases}$$
  $\mathbf{x} \leftarrow \mathbf{x}_{best} + \mathcal{N}(0, \gamma^t)$
  $d = D(f(G(\mathbf{x})), E) - D(f(G(\mathbf{x}_{best})), E)$
  if $dE < 0$ **or** $e^{-d/T} < \texttt{random}(0)$ **then**
    $\mathbf{x}_{best} \leftarrow \mathbf{x}$
  end if
end for
return $\mathbf{x}_{best}$

To improve optimization speed, at each iteration we sample a batch of multiple faces and choose the best face as the guess based on the FaceNet distance metric.

In order seed the search algorithm with an initial guess, we pre-generate a set of $160,000$ standard normal latent vectors. We run each vector through StyleGAN followed by FaceNet to produce a face embedding for each latent vector, and store both the latent and embedding vectors. Then, given a face embedding E, we can directly identify the latent vector **x** whose face embedding most closely matches the target face embedding E. Rather than initializing the search algorithm with the zero vector, we can initialize with the best identified latent vector from the pre-generated set.

Since initial stages of the search involve a long and expensive search over randomly-generated latent vectors, we can significantly speed up this initial search by pre-generating pairs of latent vectors and the corresponding FaceNet embeddings created by generating an image, aligning, normalizing, then running the result through FaceNet. Using these generated pairs, it is very fast to determine the closest embedding vector to a target embedding, and find the corresponding latent vector used to generate it. Then the rest of the search proceeds from there.

3 Experiments

We use images from the Flickr-Faces-HQ (FFHQ) data set [7]. FFHQ provides $\approx$ 70K high-quality face images of real people at 1024×1024 resolution with variation in age, ethnicity, image background, and accessories (e.g. eye-wear). These images come from Flickr, and are made publicly available under the Creative Commons BY-NC-SA 4.0 license by NVIDIA Corporation. We use a subset of 20 target images for our reconstruction algorithm, chosen at random. For each algorithm setting, we run optimization for 200 iterations with a batch size of 8. At each step, the noise standard deviation is multiplied by a factor of 0.98. We run all experiments on a Tesla P4 GPU on AWS. Reconstructing a face image from a template takes around 5 minutes.

3.1 Results

We run reconstruction on a set of 20 target images from the FFHQ data set under each of the four parameter settings. In Figure 2, we display ten of the target images, along with the reconstructed faces produced for each parameter setting. We see a significant visual improvement from the reconstructions produced using, simulated annealing, especially with the help of our pre-generated set, and in many of the faces the reconstructed face appears to closely match the target facial identity. Notably, the reconstructions are generally able to identify facial features such as hair color, eye color, and eye-wear.

`anneal`	`pregen`	L_2 Distance	Cosine Distance	Avg. # Updates
✗	✗	0.653 ± 0.110	0.213 ± 0.037	19.9
✗	✓	0.694 ± 0.132	0.231 ± 0.051	7.2
✓	✗	0.512 ± 0.077	0.165 ± 0.025	25.7
✓	✓	0.485 ± 0.081	0.156 ± 0.027	23.3

Table 1: Reconstruction quality for 20 faces from the FFHQ data set. We report mean and standard deviation (mean ± std) for L_2 distance and cosine distance with each parameter setting. We also report the number of times the candidate $\mathbf{x}_{best}$ was improved upon by a new guess, which we expect to indicate the level of exploration done during optimization.

Table 3.1 displays the average L_2 and cosine distance between the target face and the reconstructed face for each parameter setting, over the 20 target faces. Annealing and pre-generation offer an improvement by each metric, but suffers in the case of greedy optimization. Note that when using simulated annealing, the average L_2 distance falls below the 0.6 threshold, fooling FaceNet. This can be explained by the last column of the table, which shows that in the greedy and pre-generated case, the optimization stopped after very few updates, likely suggesting that the procedure got 'stuck' at a local minimum near the pre-generated face.

Figure 3: Visual comparison between our method and two previous facial reconstruction methods, Li et al. [8] and Zhmoginov et al. [14]. We note that in Zhmoginov et al., the authors assume white-box access to the facial embedding network and use a guiding image of a generic face for reconstruction.

3.2 Comparison

Next, we qualitatively compare our face reconstructions to results achieved in Li et al. [8] and Zhmoginov et al. [14]. In Figure 3, we display the target images, along with our reconstructions and those of the previous methods. Our reconstruction is more life-like in each case. Compared to Li et al., our synthesized images are higher resolution and appear to more closely preserve identity. While our images are clearer than those from Zhmoginov et al., their method picks up more fine-grained details of the target face. However, we note that in Zhmoginov et al., the authors assume white-box access to the facial embedding network and use a guiding image of a generic face for reconstruction, whereas we use a black-box method.

4 Conclusion/Future Work

We presented a method to reconstruct a person's facial identity using a face embedding generated by a facial recognition system. Our method produces reconstructed facial images which not only fool face recognition systems, but are also convincingly real to humans. The results of this work suggest the need for further review and study of facial embedding encryption systems. The code used for this project will be made available at `https://github.com/evendrow/face-reconstruction`.

References

[1] 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*, pages 4432–4441, 2019.

[2] Vishnu Naresh Boddeti. Secure face matching using fully homomorphic encryption, 2018.

[3] Forrester Cole, David Belanger, Dilip Krishnan, Aaron Sarna, Inbar Mosseri, and William T. Freeman. Synthesizing normalized faces from facial identity features, 2017.

[4] Chi Nhan Duong, Thanh-Dat Truong, Kha Gia Quach, Hung Bui, Kaushik Roy, and Khoa Luu. Vec2face: Unveil human faces from their blackbox features in face recognition, 2020.

[5] Marta Gomez-Barrero and Javier Galbally. Reversing the irreversible: A survey on inverse biometrics. *Computers & Security*, 90:101700, 2020.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] 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*, pages 4401–4410, 2019.

[8] Zhigang Li and Yupin Luo. Generate identity-preserving faces by generative adversarial networks, 2017.

[9] Guangcan Mai, Kai Cao, Pong C Yuen, and Anil K Jain. On the reconstruction of face images from deep face templates. *IEEE transactions on pattern analysis and machine intelligence*, 41(5):1188–1202, 2018.

[10] K. Nandakumar and A. K. Jain. Biometric template protection: Bridging the performance gap between theory and practice. *IEEE Signal Processing Magazine*, 32(5):88–100, 2015.

[11] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[12] Peter JM Van Laarhoven and Emile HL Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer, 1987.

[13] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 225–240, New York, NY, USA, 2019. ACM.

[14] Andrey Zhmoginov and Mark Sandler. Inverting face embeddings with convolutional neural networks. *arXiv preprint arXiv:1606.04189*, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] As we suggest, our method is able to produce life-like reconstructions of a target. See Figure 2 for a visualization and Figure 3 for a comparison to previous works.
   (b) Did you describe the limitations of your work? [Yes] We describe the limitations of our experiments, specifically that we only apply our method to a small subset of the FFHQ data set.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Yes, We describe how our method and similar ones can be used by attacks to obtain personal identity information from users of biometric authentication systems.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] A URL to the accompanying github repository is included.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Experimental details are provided in Section 3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Table 3.1 provides standard error for face embedding distance.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Information is provided in the Experiments section.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite StyleGAN, FaceNet, and FFHQ.
   (b) Did you mention the license of the assets? [Yes] We state the license for FFHQ 3.
   (c) Did you include any new assets either in the supplemenmaterial or as a URL? [Yes] A URL to the accompanying github repository is provided
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We state that the images we use are publicly available in 3.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Yes, we mention in 3 that these images come from real people.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]