IdDecoder: A Face Embedding Inversion Tool and its Privacy and Security Implications on Facial Recognition Systems

Minh-Ha Le
Linköping University
Sweden

Niklas Carlsson
Linköping University
Sweden

ABSTRACT

Most state-of-the-art facial recognition systems (FRS:s) use face embeddings. In this paper, we present the IdDecoder framework, capable of effectively synthesizing realistic-neutralized face images from face embeddings, and two effective attacks on state-of-the-art facial recognition models using embeddings. The first attack is a black-box version of a model inversion attack that allows the attacker to reconstruct a realistic face image that is both visually and numerically (as determined by the FRS:s) recognized as the same identity as the original face used to create a given face embedding. This attack raises significant privacy concerns regarding the membership of the gallery dataset of these systems and highlights the importance of both the people designing and deploying FRS:s paying greater attention to the protection of the face embeddings than currently done. The second attack is a novel attack that performs the model inversion, so to instead create the face of an alternative identity that is visually different from the original identity but has close identity distance (ensuring that it is recognized as being of the same identity). This attack increases the attacked system's false acceptance rate and raises significant security concerns. Finally, we use IdDecoder to visualize, evaluate, and provide insights into differences between three state-of-the-art facial embedding models.

CCS CONCEPTS

• **Security and privacy**; • **Applied computing**; • **Computing methodologies** → **Machine learning**; **Computer vision**;

KEYWORDS

Face embedding inversion, Black-box attack, Facial recognition

ACM Reference Format:
Minh-Ha Le and Niklas Carlsson. 2023. IdDecoder: A Face Embedding Inversion Tool and its Privacy and Security Implications on Facial Recognition Systems. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy (CODASPY '23), April 24–26, 2023, Charlotte, NC, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3577923.3583645

1 INTRODUCTION

Advances in deep learning, and Convolutional Neural Networks (CNNs) in particular, have helped push the state-of-the art in face recognition. Today, most state-of-the-art facial recognition systems (FRS:s) use models that embed facial images to low dimensional embeddings (usually 128 or 512 dimensions). This approach has several advantages. First, the use of embeddings provides a robust and lightweight way to cluster and recognize an identity without the need to access the real image dataset. Second, it avoids having to store privacy-sensitive biometric data such as facial images.

Given these advantages, it is not surprising that the top performing (in terms of accuracy), recently proposed models are embedding models [1–3]. However, face embeddings are currently not considered confidential and most models do not conceal their embeddings. Furthermore, there is pressure on many organizations to digitize and move their services to the cloud. Combined, based on these observations, we foresee an increasing number of embeddings to be easily obtained by attackers. For example, an attacker may learn about embeddings due to poor protection when the embeddings are at rest (e.g., the stored embeddings associated with a gallery set) or in transit (e.g., the remote database lookups when checking whether a person is in the galley set). With the embeddings corresponding to facial representations, one of the most clear personal identifiers, it is therefore important to ask (1) *whether and to what degree exposed face embeddings can reveal privacy-sensitive information.* For example, is it possible to build a model that can reveal the identity (or face) of a person given an exposed embedding from the gallery set? If so, an attacker that gains access to the embeddings of a gallery set could easily expose all registered identities (and hence also group membership). Furthermore, non-protected or weakly protected database lookups against cloud-hosted databases (e.g., criminal database used by the police) can reveal people of interest to whoever performs the lookup.

Here, it should be noted that several embedding APIs are publicly open (e.g., Clarifai, Microsoft face API, Rekognition, etc.) and many pre-trained embedding models (e.g., FaceNet, Dlib, ArcFace) are easily accessible on the Internet. To further motivate the importance of this question, we note that, at the time of writing, Clarifai [4] provides an API that allows users to submit face images and receive back a face embedding in plaintext (e.g., see police use case above).

Assuming successful inversion, it is also important to ask (2) *to what degree alternative identities can be created that tricks FRS:s using embeddings in new ways that weakens their security and reputation.*

In this paper, we present the IdDecoder framework and two novel attacks on state-of-the-art facial recognition models using embeddings. Both attacks incorporate IdDecoder in their design and together address questions (1) and (2). The first attack, a new variant of *model inversion* attack [5], is shown to successfully synthesize a face capturing the identity of the person from its corresponding embedding (exposing their identity to the attacker). The second one is a novel attack against the FRS called a *false acceptance* attack. This attack provides the attacker with a tool to create synthesized faces of different identities that trick the attacked FRS to believe that the different identities are the same as the original identity

CODASPY '23, April 24–26, 2023, Charlotte, NC, USA

ACM ISBN 979-8-4007-0067-5/23/04.
https://doi.org/10.1145/3577923.3583645

despite clearly depicting other people (significantly reducing the systems accuracy). The first attack raises privacy risks with state-of-the-art FRS and demonstrates the importance of protecting face embeddings. The second attack demonstrates security risks for systems using such FRS, as they may be tricked to wrongly accept a different identity. We also use IdDecoder to provide visual (and numeric) insights into the relative efficiency of different state-of-the-art models under attack. The idea behind IdDecoder is derived from the first presented attack, while the other contributions are derived via desirable properties inherited from our design of IdDecoder.

Model Inversion (MI) attack: In 2015, Fredrikson et al. [5] demonstrated a black-box-based model inversion attack against a facial recognition API, where the attacker can submit images to the API and receives back a confidence value on a specific or the whole range of labels (classes). In their work, the attacker's goal is to reconstruct the training images corresponding to the label, raising concern that privacy-sensitive information in the training dataset is leaked. However, given the limited information provided by the confidence values, the reconstructed images are usually blurred, and the human eye is typically not able to recognize the same identity. Others [5–8] have since proposed MI attacks that make use of white-box settings, auxiliary information, or both to improve the results. Most of these attacks (cf. Table 2) are against classification models that are simpler and less accurate than the current state-of-the-art solutions using face embeddings (considered here).

Here, we present a model inversion attack against facial recognition models using face embeddings that achieves more realistic results than prior work (see Section 5) despite operating in a black-box setting and without access to any auxiliary information. Our attack is a black-box attack in the sense that the attacker does not have access to the parameters but instead assumes that the attacker can obtain the embeddings generated by the model.

Given the model and one or more embeddings of interest, the goal of the attacker is to decode the face embedding and reconstruct a realistic face image that is visually recognized as the same identity as the original face used to create the embedding(s). We realize this goal through the introduction of a mapper (that we train) and an optimizer (that does not need training but benefits from output from the mapper). These two components are at the heart of our IdDecoder framework and allow us to recreate a synthesized face with the same identity and high visual resemblance as the face corresponding to any given embedding associated with the attacked model. The power and generality of IdDecoder is demonstrated through evaluation on several state-of-the-art embedding models, including FaceNet [1], ArcFace [2], CurricularFace [3], and Dlib [9].

False Acceptance (FA) attack: We next propose a false acceptance (FA) attack that undermines the FRS by increasing its false acceptance rate. Assuming knowledge of a face embedding, the attack is achieved by the attacker generating face images of a different identity than the victim identity but that has sufficiently similar embedding as the original identity that the FRS considers them to be of the same identity. We again leverage IdDecoder in the design of this attack but make a change to the objective function of the model so to generate different faces that belong to the same embedding. The main differences compared to the MI attack is that the FA attack generates a face that visually belongs to a different identity, but the FRS is fooled to classify them as the same identity.

The efficiency of our FA attack highlights a major weakness in current state-of-the-art FRS:s, allowing an attacker to significantly reduce the security and reputation of deployed FRS:s. While having received much less attention than the MI attack, our results show the need to consider FA attacks in the design of future FRS:s.

Summary of contributions: (1) We propose a novel face embedding decoder framework, IdDecoder, that effectively can synthesize realistic-neutralized face images from face embeddings. (2) We use IdDecoder to perform an effective black-box model inversion attack against state-of-the-art facial recognition models. (3) We present a false acceptance attack against facial recognition models, demonstrate how it can be implemented using IdDecoder, and show that it can significantly affect the accuracy and security of facial recognition models. (4) We us IdDecoder to visualize and provide insights into the effectiveness of different facial embedding models.

Outline: Section 2 describes the problem and threat models considered. After a technical background to GANs in Section 3, Section 4 presents the design of the IdDecoder framework and the specific attacks that it enables. Evaluation results are presented in Section 5. Sections 6 and 7 present related work and conclusions.

2 PROBLEM AND THREAT MODELS

We focus on FRS:s using embeddings. These embeddings can be seen as learned representations of facial features.

2.1 Facial Recognition

The two main approaches to building a FRS using CNNs are to build: (1) a multiple-class classifier [10–12] in which each class is an identity, and (2) a representational model [1–3, 9] that learns a lower-dimensional face embedding from the face images. While the multi-class approach typically achieves high accuracy using standard benchmark such as LFW [13], the complexity of these models increases substantially with the number of identities in the system and they do not work well for open-set facial recognition [2]. Due to its higher accuracy and greater flexibility, the embedding approach has recently become more popular.

In addition to superior facial recognition performance, the use of embeddings can improve efficiency, reduce the need to store facial images sets (only the embeddings must be stored), and they make it easier to scale up the tasks of face verification and face clustering with open-face datasets (i.e., datasets not used for training).

2.2 Training, Gallery, Probe Set

At least three types of datasets are used by a FRS based on embeddings: (1) The training dataset $\mathbb{T}$ is used for training the embedding model. This set can include both evaluation and testing sets used to learn the unique identity features in each face (i.e., the face embedding). For simplicity, we refer to the combined set as the training set. (2) The gallery set $\mathbb{G}$ includes the set of known faces (registered faces). This set may include many subjects not in the training set $\mathbb{T}$. Yet, the system must be able to extract identity features uniquely from all the registered faces. (3) Finally, the probe set $\mathbb{P}$ includes the set of faces that the system should try to find matches for in $\mathbb{G}$. Also the subjects in the probe set $\mathbb{P}$ might differ from the two prior sets, and may therefore include faces totally new to the FRS.

Our attacks at a glance: We present two attacks against FRS:s using embeddings. In the *model inversion (MI) attack*, we try to learn faces from the gallery set given only access to the embeddings. In the *false acceptance (FA) attack*, we again target the gallery set. However, in this case we try to create new face images of a different identity but that have sufficiently similar embedding that the two images (with different identities) are classified as the same person.

2.3 Embedding-based Facial Recognition

The face embedding model $\mathcal{M}$ can be seen as a deterministic function $\mathcal{M} : \mathbb{R}^{H\times W\times D} \mapsto \mathbb{F}^d$ that maps from the pixel space of an input face image (where H, W, and D are the image height, width, and depth) to an identity feature space of dimension d. Let $\mathcal{G} = \{G_0, G_1, ..., G_{m-1}\}$ be a gallery set with m registered faces and $\mathcal{F}_g = \mathcal{M}(\mathcal{G}) = \{\mathcal{M}(G_0), \mathcal{M}(G_1), ..., \mathcal{M}(G_{m-1})\} = \{g_0, g_1, ..., g_{m-1}\}$ be the embedding representation of these faces in $\mathcal{G}$, where $\mathcal{F}_g \in \mathbb{F}$. Then, given a probe set $\mathcal{P} = \{P_0, P_1, ..., P_{n-1}\}$ with n face images, the face embedding $p_i = \mathcal{M}(P_i)$ is first extracted for each face P_i ($0 \le i < n$). Second, each such embedding p_i is compared against each face embedding in the set of embeddings $\mathcal{F}_g$ of the gallery set so to obtain a ranking order of the k closest faces of each image P_i. We denote the top-k list for image P_i as $R_i = \text{rank}^k_{\mathcal{F}_g}(p_i) = (r_1, r_2, ..., r_k)$.

2.4 Threat Models for the Two Attacks

Face embedding MI attack: The traditional MI attacks [5, 7, 14] focus on black-box settings (adversary is given access to the API) or white-box setting (given full access to the model) and use predictions on chosen labels to extract sensitive features in the training set $\mathbb{T}$ (equal to the gallery set $\mathbb{G}$ in the case of [5]). These prior attacks specifically target classifier models. In contrast, our MI attack targets FRS:s using embeddings. Specifically, we consider an adversary who intercepts and captures the face embeddings and uses these to reconstruct the corresponding faces. The attack assumes that the attacker can query the (black-box) facial recognition API for face embeddings of submitted faces but that it may not have access to the whole gallery/training dataset. Given an embedding, we show that the attacker can reconstruct a clear and realistic face with high success rate and resemble almost exactly the identity under attack. Compared to the above discussed related works [5, 7, 14] (targeting classifier-based models), our results provide much better identity resemblance and is applicable to more modern systems.

In addition, our attack is both feasible and practical. First, several face embedding APIs are publicly open (e.g., Clarifai, Microsoft face API, Rekognition, etc.) and the many pre-trained embedding models (e.g., FaceNet, Dlib, ArcFace) are easily accessible on the Internet. Furthermore, at the time of writing, the face embeddings are not considered confidential or necessary to conceal (e.g., using encryption). Instead, APIs such as Clarifai [4] have an option to return the face embedding via their API.

FA attack: In the second attack, we consider an attacker who seeks to undermine the FRS by increasing its false acceptance rate. In the simplest case, involving a single identity, the attacker is assumed to have access to the face embedding of the victim's identity. (This embedding can either be obtained in the same way as for the MI attack, or more generally, the attacker can extract the victim's face embedding from almost any face images available via social media, CCTV cameras, or collected any other way). Using this embedding, the goal of the attacker is to generate a non-identical clone of the victim's face with the same or very similar embedding. By non-identical clone we mean that the generated face looks sufficiently different than the original face that a human easily distinguished it as belonging to a different identity, even though the FRS classifies it as belonging to the same identity. By generating many such images, the attacker can substantially increase the systems false acceptance rate. To avoid (or get around) the subjective nature of human evaluation, we make creative use of different face embedding models in the design of this attack. Specifically, the attacker creates an optimal clone that passes the verification threshold of the FRS under attack (say system A) while it stays well below the verification threshold of another FRS (say B). Using the systems against each other this way, we show that the attacker typically can find vulnerabilities (or gaps) in system A (with the help of system B) that can be used to generate false positives.

While we are only concerned with 2D images here, the attack naturally expands to 3D. In this case, the attack could also be used to break state-of-the-art face authentication such as face ID on iPhones and Androids. While these authentication systems incorporate the use of 3D-depth sensors (for higher security), the fact that these systems rely only on well-matching embeddings combined with our approach's ability to identify an alternative identity with the same (or very similar) embeddings, but for a different identity, suggests that a 3D model of a face that produce the same embedding can be created. With the help of a 3D printer the attacker can then print a 3D clone face that passes the facial authentication systems.

3 TECHNICAL BACKGROUND ON GANS

Our IdDecoder relies heavily on the use of GANs. Here, we provide the technical background and notation used to describe IdDecoder.

During the training phase, a GAN model aims to learn the data distribution of the original training data. Once trained, the model can randomly generate samples from this distribution.

More technically, without loss of generality, we can consider some input data with N samples $\{I_1, I_2, ..., I_N\}$. Typically, the samples $I \in \mathbb{R}^q$ are high-dimensional data (e.g., images, voices, text encodings) assumed to be randomly sampled from a distribution P_I; i.e., $I_i \sim P_I$. Given this, the objective of a GAN is to learn a generative model that can produce samples from P_I.

During training, GANs use a special training workflow called adversarial learning. Here, two entities, the generator G (which generates new samples) and the discriminator D (which tries to distinguish the generator's samples as real or fake), are trained adversarially, while gradually helping each other to improve. More specifically, G takes random noise vectors $z \in \mathcal{Z}$ of dimension $p \ll q$, where $\mathcal{Z} = \mathbb{R}^p$ as inputs, and generates a new sample I with the expectation that $I \sim P_I$. In contrast, the discriminator D is a classifier network that takes a sample I as input and classifies it as real or fake, $D : \mathbb{R}^q \mapsto [0, 1]$.

One of the important advantages of using a GAN on image data is its highly disentangled latent space $\mathcal{W} = \mathbb{R}^p$, which is an intermediate space between the latent space $\mathcal{Z}$ and sample space $\mathbb{R}^q$. Slightly different from the canonical GAN [15] described above, the state-of-the-art StyleGAN models for images [16–18]

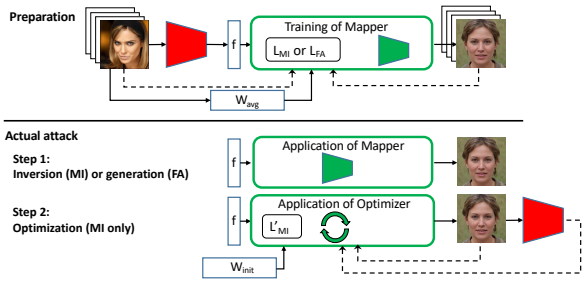


Figure 1: Overview of the IdDecoder framework both during training and execution of the two different attacks.

introduce the latent space $\mathcal{W}$ and map the vector $z \in \mathcal{Z}$ to latent codes $w \in \mathcal{W}$. The w vector is replicated as a layered array in $\mathcal{W}+$ of size $p \times l$, where l is number of layers, and then passed through different layers of the pyramid structure of the generator G to produce the sample I. Empirical results have shown that the latent spaces $\mathcal{W}$ and $\mathcal{W}+$ are more disentangled compared to $\mathcal{Z}$. Specifically, different layers of the latent codes w (and $w+$) typically are responsible for different visual attributes of the image sample. For example, with facial images, the vector z typically has a regular size $p = 512$ and a latent codes w has $l = 18$ layers. Hence, w has size (512×18). Furthermore, Karras et al. [16] have shown that the coarse layers 1-to-4 represent coarse spatial resolutions such as pose, general hair style, and face shape; the medium layers 5-to-8 are responsible for smaller-scale facial features such as hair style, eyes open/closed; and the fine layers 9-to-18 represents the color scheme and microstructures observed in a facial image.

4 IDDECODER FRAMEWORK

In this section, we present the IdDecoder framework and the loss functions used during training and optimization.

4.1 Framework Overview

Fig. 1 presents an overview of the IdDecoder framework. In the preparation phase (top row), the mapper (or as we will see a mapper network) is trained using a catalogue of images and one out of two loss functions. One for the MI attack and one for the FA attack. Here, the main differences is that for the MI attack we design the loss function L_{MI} to help reconstruct the face of a given face embedding, whereas for the case of the FA attack we relax some parts of the loss function to allow generation of a non-identical clone with L_{FA}. During the actual attack, the trained mapper can then be used to very efficiently generate an inverted face (MI attack) or a face of an alternative identity than the origin face (FA attack). For the MI attack, our framework also adds an optimization step in which through a series of optimization steps the results is optimized with the help of the attacked model, the output from the previous step, and a loss function L'_{MI} (same as L_{MI} but with different hyperparameter values than with the mapper).

High-level mapper: Face recognition models using embeddings compress the information of images into a small embedding $f \in \mathbb{R}^d$, where d is the system-dependent embedding size. For example, FaceNet takes a normalized RGB face image of size 160×160 and extracts a face embedding of size d=128. This corresponds to a compression ratio of 600 (3×160×160/128). To overcome the challenge

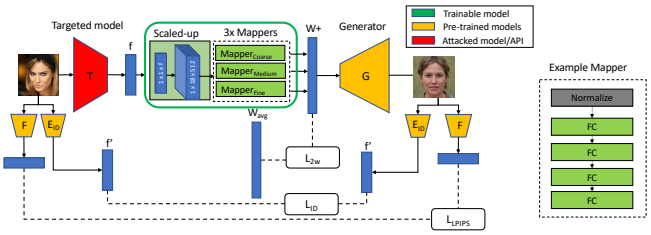


Figure 2: Overview of the mapper used in the framework.

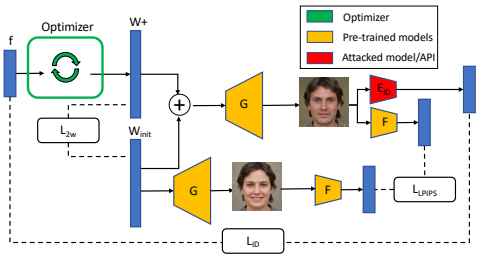


Figure 3: Overview of the optimizer used in the framework.

of reconstructing a face from the much smaller embedding f, IdDecoder uses a mapper network $\mathcal{M}$ that maps f onto the latent space $\mathcal{W}+$ of StyleGAN. ($\mathcal{W}+$ has been shown to hold more information than $\mathcal{W}$ [19].) The key idea here is that, given a good mapper, these latent codes $w \in \mathcal{W}+$ then can be used to regenerate a facial image R using a pre-trained StyleGAN generator G.

Fig. 2 presents an overview of the mapper used in IdDecoder. The framework takes a black-box target model T as input (red cone), which itself takes a facial image I as input to create a face embedding f as output. Then, as described above, a mapper network $\mathcal{M}$ (green cone in Fig. 1 or green box in Fig. 2) takes the embedding f as input and outputs a latent code $w \in \mathcal{W}+$ that we can use to reconstruct a facial image R using a pre-trained StyleGAN generator G. Here, yellow cones indicate the use of pre-trained models.

In summary, assuming that we have properly trained the mapper $\mathcal{M}$, we can now reconstruct an input image I with face embedding $f=T(I)$ by simply passing the facial embedding through the mapper and then pass the resulting latent code $w=\mathcal{M}(f)$ through G to obtain $R=G(w)$. The main challenges to achieve good results lie in the design and training of the mapper. Here, we make use of several important insights.

4.2 Mapper Abstraction in More Detail

Let us look closer at the steps taking us from embedding f to regenerated image R in more detail.

First, the input to the mapper(s) requires the face embedding $f \in \mathbb{R}^d$ to be scaled up to the sample size of the latent codes of (512×18). Here, we use a fully connected multilayer perceptron (MLP) with Leaky ReLU activation function as scaled-up network, where the input size d depends on the target model T. In particular, d=128 for FaceNet [1] and dlib [9], d=512 for ArcFace [2] and CurricularFace [3], and d=1024 for Clarifai [4].

Second, and perhaps most importantly, we use three different mappers, each responsible for their own set of layers: one for the coarse layers (1-4), one for the medium layers (5-8), and one for the fine layers (9-18). We call the corresponding mappers "coarse",

"medium", and "fine", respectively. This design and the naming of the layers were motivated by prior observations that different layers in the latent codes w (used as input to G) are responsible for different feature sets [16]. Through experiments we have found this design (with separate mappers for different layers) highly beneficial compared to using a single mapper, as it allows each mapper to focus on specific features. More specifically, each mapper is implemented as three smaller fully connected neural networks (illustrated to the right in Fig. 2), each with four fully connected (FC) layers, and each of them is a MLP with Leaky ReLU activation function.

Finally, the latent code w is passed to a pre-trained StyleGAN model G (trained on the FFHQ dataset [16]) that generates an output image of resolution 1024×1024.

4.3 Training of Mapper(s)

During the training phase, we make use of several loss functions that either extract information from the input images $\mathcal{I}$ and the corresponding regenerated image R of each image $I \in \mathcal{I}$, or in latent space. Referring to Fig. 2 we include two functions of the first type (L_{ID} and L_{LPIPS}) and one of the second type (L_{2w}).

First, and perhaps most importantly we calculate the identity loss L_{ID} as the pairwise distance

$$L_{ID} = \|E_{ID}(I) - E_{ID}(R)\|_2 \tag{1}$$

between the face embeddings of the original face I and the reconstructed face R, where the embeddings are calculated using a pre-trained face embedding model E_{ID} capturing identity features of a face image. Note that E_{ID} does not have to be (although it can be) the same model as the model under attack (i.e., T).

The identity loss helps ensure that the identity features in I and R are the same. However, it does not provide any guarantee that the reconstructed image R is a facial image. To tackle the challenge of producing a realistic image, we use the pairwise loss in latent space L_{2w} and the perceptual loss L_{LPIPS} [20] in pixel space. L_{2w} is calculated as the pairwise distance in latent space in which it constraints the output W close to the average latent W_{avg}:

$$L_{2w} = \|W - W_{avg}\|_2. \tag{2}$$

Finally, L_{LPIPS} is calculated based on a pre-trained perceptual feature extractor F, again taking the pairwise distance:

$$L_{LPIPS} = \|F(I) - F(R)\|_2. \tag{3}$$

The choice to use the pre-trained perceptual feature extractors LPIPS is motivated by works [17, 21] having shown that it outperforms standard perceptual losses [16, 22]. We next describe how the above loss functions and modifications of them are used to achieve the MI and FA attack, respectively.

4.4 Attack-specific Loss Functions

Face reconstruction in MI attack: The goal when training the mapper $\mathcal{M}$ for use in the MI attack is to learn to construct an output latent code $w = \mathcal{M}(E_{ID}(I))$. For reconstruction we give positive weight to all three losses. In particular, the total loss is defined as:

$$L(I, R) = \lambda_1 L_{ID} + \lambda_2 L_{2w} + \lambda_3 L_{LPIPS} \tag{4}$$

where λ_1, λ_2, λ_3 are hyperparameters that can be tuned.

Construction of non-identical clones in FA attack: For this attack, the objective is to generate a non-identical clone R of I to fool the facial recognition to accept the clone as being of the same identity. Different from the MI attack, our goal when training the mapper for the FA attack is to learn to construct an output that satisfy $W = \mathcal{M}(E_{ID}(I)) + W_{avg}$. In other words, we train the mapper in the way that it learns the difference $\Delta = W - W_{avg}$ to best satisfy the losses L_{ID}, L_{LPIPS} and L_{2w}. This helps direct the mapper to learn how to project a semantic representation onto latent space that can allow the reconstruction of an image with similar identity distance but that avoids identity-specific details to be enforced. More importantly, we found that the mapper can learn to project an identity representation that is not visible to the human eye but that still ensures that the identity distance of the constructed image is small. This allows us to project the non-visible identity of any attacked face on to a face image.

However, a technical challenge with this method is that it is easy that the mapper converges on some random areas in latent space $W+$. As a result, the reconstructed images become visually similar. To address this problem, we introduce the loss function L_{INNER}, which is the L2 loss between a randomly chosen pair of samples in the mapper's output batch. Assuming that each training batch of the mapper $\mathcal{M}$ has $m \geq 2$ samples $\{f_0, f_1, ..., f_m\}$, the mapper outputs $\{w_0, w_1, ..., w_m\}$, where $w_i = \mathcal{M}(f_i)$, $iin[0, m]$. We can then define

$$L_{INNER} = \|w_p - w_q\|_2, \tag{5}$$

where $p, q \in [0, m]$. Given this additional loss term, the modified loss function (used by the FA attack) reads as:

$$L(I, R) = \lambda_1 L_{ID} + \lambda_2 L_{2w} + \lambda_3 L_{LPIPS} + \lambda_4 L_{INNER}. \tag{6}$$

where λ_1, λ_2, λ_3 and λ_4 are hyperparameters that can be tuned.

4.5 Optimizer (used in MI Attack only)

The optimizer is shown in Fig. 3. In contrast to the mapper (Fig. 2), the optimizer does not require any training. This has many advantages. For example, the whole optimization process typically completes after only 20-40 iterations; hence, it typically only requires 20-40 API queries to the attacked model E_{ID} per attacked embedding f. This is attractive in the case we only have access to a limited number of queries to E_{ID}. Moreover, we can easily swap different target models E_{ID} without any complication. In fact, compared to the mapper, the optimizer is better reconstructing the unique appearance of a particular person. Starting from W_{init}, the optimizer runs back propagation [23] with the loss function in equation (4) (with a different hyperparameter set than the mapper).

5 EXPERIMENTS AND EVALUATION

5.1 Datasets and Implement Details

For training and evaluation, we use the following target models: ArcFace [2], FaceNet [1], CurrricularFace [3]. For training we use the images from CelebA-HQ [24]. The CelebA-HQ dataset [24] has 30,000 high-resolution face images sampled from the CelebA [25] dataset, containing images of approximately 8,000 identities. For evaluation, we use the UTK dataset [26], CelebA, and the Labelled Faces in the Wild (LFW) dataset [13]. The LFW dataset contains 5,749 identities and provides a good reference point as most FRS:s

use this dataset for evaluation (not training). The UTK dataset contains more than 20,000 images labelled with age, gender, and ethnicity. We use this dataset to evaluate biases.

Our training process is divided into two phases. First, we train the model for 30,000 steps with learning rate 0.1. The high learning rate helps the model finding the global minimum better, while avoiding getting stuck on local minimum. Second, we reduce the learning rate 0.1 and keep training for another 30,000 steps. This training strategy helps reduce L_{ID} from the first phase and improves the image quality by reducing artifacts during the second phase. For the MI attack, we set loss parameters as follows: $\lambda_1 = 0.2$, $\lambda_2 = 0.8$, $\lambda_3 = 0.8$. For FA attack, we set $\lambda_1 = 0.2$, $\lambda_2 = 0.8$, $\lambda_3 = 0.01$, and $\lambda_4 = 0.005$. In both of the attacks we recommend setting $\lambda_1 \leq 0.2$ due to the fluctuation of the L_{ID} early in the training process. To find these settings, we started with empirical settings based on prior successful models and then iteratively fine-tuned the parameters.

For the evaluation, we simply calculate the face embedding f for each image $I \in \mathcal{I}$ in the evaluation set and then compare it with the corresponding reconstructed face R of each such embedding. For the reconstruction, we use the mapper, the optimizer, or both.

5.2 Evaluation Metrics

Perhaps the most important aspect of both attacks considered is the identity distance to the original identity. While we also use an identity distance in the loss function used for training our mappers (e.g., equations (4), (6)), it is important to note that we can use any embedding for this purpose. For fair comparison, we do not use the same face embedding model E_{ID} as we use in the training process of our mappers. Instead, unless explicitly stated, we will use the targeted face embedding model T to evaluate how effective the IdDecoder is against T. This choice captures that these state-of-the-art FRS:s (based on face embedding) make their identity identification decision based on whether two images are within an identity distance threshold from each other. Here, we calculate the identity distance between the reconstructed face R and the ground truth face I and determine whether the pairwise distance is below the identification threshold. By repeating this for many images (i.e., all identities in LFW) we can also report the percentage of reconstructed faces that pass the identification threshold.

While the identities in LFW may be different than the identities of the gallery sets $\mathcal{G}$ of the actual FRS:s that we evaluate, we believe that the LFW dataset provides a good sample set for this evaluation. Finally, we note that in the case of the MI attack the attacker would only have access to $f = T(I)$ (not I) and we only give IdDecoder the embedding f as input when performing an attack. Image I is therefore only used for evaluation purposes.

When comparing different FRS:s, the best choice of identification threshold differ between models. In general, a more relaxed threshold increases the classification rates but also the false acceptance rate (FAR) and the best choice typically depends on the embedding size d (e.g., 128, 512 or 1024, etc.). Since regular normalization methods (e.g., linear scaling, clipping, or log scaling) reduce the accuracy of the FRS, we base the threshold choices on the results and discussions provided by the authors of the original models [13].

Beside the identity distance, we also use three other metrics to measure the similarity and distance between the reconstructed images to their ground truth: Structural Similarity Index (SSIM) [27], SSIM's enhanced version Multiscale SSIM (MS-SSIM) [28], and LPIPS [20]. These metrics try to imitate the human visual perception where the structural information is used to compared between images. For example, SSIM extracts three key features, including luminance, contrast, and structure from the image and using these features when comparing two images. MS-SSIM is a variation of SSIM, where the measurements are conducted by using multiple scales of sub-sampling. The metric is as good as SSIM or better on different kinds of images or videos. Finally, LPIPS is the most advance metric in which highly convolutional networks such as AlexNet [29], ResNet [30], and VGG [31] are used to extract features at first, before those features are used to calculate the distance/similarity index. To obtain distance-based versions of the similarity metrics SSIM and MS-SSIM, we take one minus the similarity values, and refer to the resulting metrics as the distance versions of SSIM and MS-SSIM.

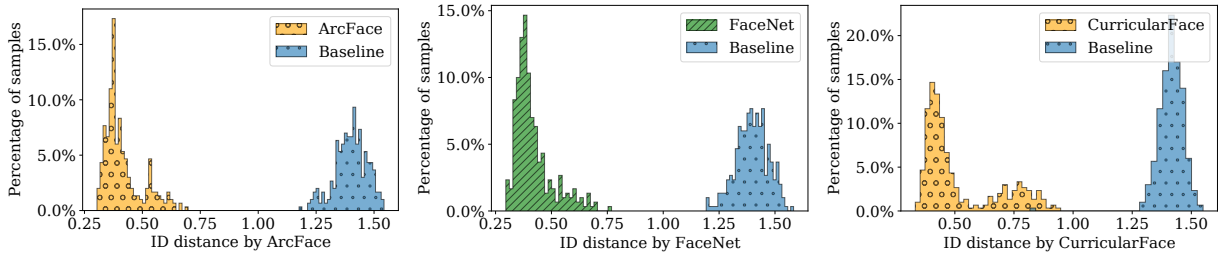
5.3 Model Inversion (MI) Attack

We next evaluate the performance of our attack against different FRS:s. As described above, for each system under attack, we trained/optimized IdDecoder using the model under attack (using our training dataset) and then used the embeddings from our evaluation dataset to evaluate the success of the attack.

Pairwise identity distances: We first show that the attack produces images that resemble the ground truth identity associated with the targeted embedding much more than a random identity and that the identity distance compared to the recognition thresholds of the systems almost always consider the generated faces as being of the same identity. This is illustrated in Fig. 4. Here, we show a frequency histogram of the pairwise identity distances between the ground truth image I and either (1) the corresponding reconstructed images R or (2) a random face in the ground truth dataset. It is important to note that the random face distributions nicely match the thresholds reported in previous work using the same evaluation dataset (referred to as ground truth in this subsection).

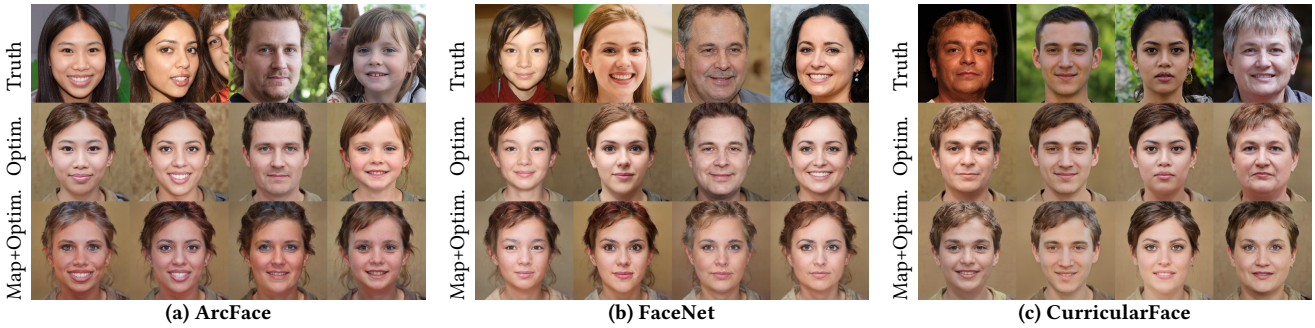
As desired, for all three systems under attack, we are able to generate images with much lower pairwise distance to the ground truth than random (see clear separation between distributions) and the majority of the pairwise identity distances typically are well below the identification thresholds of each model (discussed above).

The most successful attack is against ArcFace. With ArcFace, there is a clear separation between the distributions and the whole distribution sits well to the left of the 1.25 threshold. ArcFace would therefore recognize 100% of the reconstructed faces as being an image of the same identity as seen in the ground truth. We observe similar trends but with slight shifts toward the baseline when comparing the reconstructed distributions and the baseline for both FaceNet and CurricularFace. However, also here the distributions are widely separated from the baseline. The results show that our reconstructed faces provide face embeddings that are highly recognized as the original faces by the FRS:s even when using other image sets than those used during training.



(a) ArcFace (b) FaceNet (c) CurricularFace

Figure 4: Comparing distance distributions of the recreated images using each attacked model (ArcFace, FaceNet, CurricularFace) against the baseline of random pairs selection. In each case, we use the distances calculated using the model under attack.



(a) ArcFace (b) FaceNet (c) CurricularFace

Figure 5: Example results inverting embeddings of ArcFace, FaceNet and CurricularFace: ground truth images (1st row), reconstructed images by the optimizer (2nd row) and mapper + optimizer (3rd row).

Insights from visual example results: We next consider the visual results associated with each model. Figs. 5a-5c show a few samples of the reconstructed faces using the Optimizer (alone) and using the Optimizer in combination with the Mapper against ArcFace, FaceNet and CurricularFace. First, it should be noted that FaceNet uses a significantly smaller embedding size ($d = 128$) compared to the other two models ($d = 512$). It is therefore easy to expect that the visual results would be the least natural looking for FaceNet. However, this is not necessarily the case as the systems may use different degrees of normalization and cropping of non-identity related areas. We can clearly see that all faces are normalized in that all reconstructed identities are looking forward and do not wear lipstick, accessories, or have facial hair. This has advantages when doing facial recognition. Regarding the identity resemblance, the Optimizer appears to outperform the Mapper + Optimizer as the results look more like the ground truth identities. Interestingly, the Optimizer is capable of reconstructing the hairstyles in some cases (even though the information might be cropped out by the facial recognition). Combining the Mapper+Optimizer has other advantages. First, the combination achieves better normalization effect, where hairstyles, color scheme, facial expressions are more consistent. Most importantly, in term of performance, the Optimizer takes us 10-15 seconds per output, which is double the time that the option of Mapper+Optimizer requires (after training the Mapper).

5.4 FA Attack

The FA attack aims to create a facial image of a different identity than the victim but that would have an identity distance (defined by the attacked model) that is below the identification threshold.

Visual example results: Fig. 6 shows examples of non-identical clones for four example identities (top row) when attacking ArcFace (2nd row), FaceNet (3rd row) and CurricularFace (4th row). In all cases, the clones would have an identity distance to the original embeddings so that they would be considered to be of the same identity. Yet, comparing the images between the rows it is clear that many identity-related properties of the non-identical clones (rows 2-4) are completely different than for the original people (top row). This example shows that our system can create facial images of different identities that the facial recognition system FRS still deem to be of the same identity as the original faces. Here, we have used the average face as the baseline face but note that other faces also can be used. Being able to create non-identical clones of an attacked identity (top row) that to a human look more like an average face (or some other baseline face) can have serious security consequences for systems that use such models for authentication. Compared to the MI attack, we observe more visual artifacts and the facial appearances are more homogeneous. In particular, the mapper aims to create the best-looking images that push the boundary where human perception and the perception of the FRS:s clearly are in contrast. As seen here this comes at the cost of lower image quality.

Quantitative comparisons: As visual comparisons can be subjective, we next use quantitative metrics to provide more insights into the tradeoffs made here. First, in Fig. 7, we show the effect of the FA attack on the FRS:s. The experiment is carried out on the LFW benchmark [32]. The benchmark has a dataset containing around 13,000 facial images. The benchmarking protocol is set up as follows: (1) 12,000 samples of the dataset are selected in the way that there are 6,000 pairs in which haft of the pairs are matched identities and the other half consists of mismatched identities. (2)

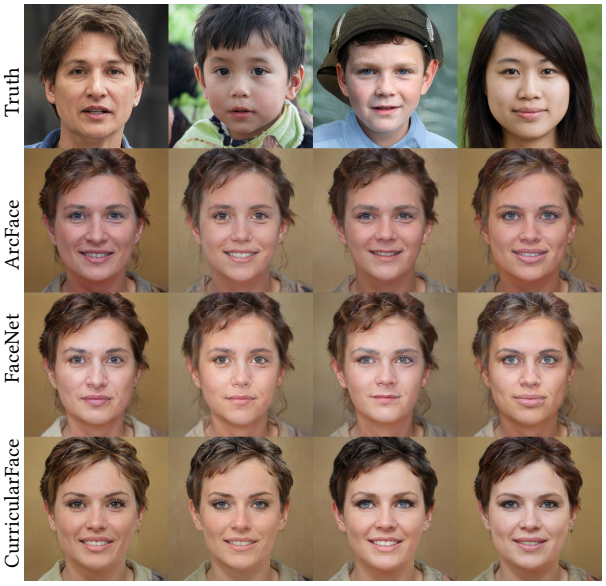


Figure 6: Non-identical clone: First row is ground truth image, and reconstructed image based on embedding of ArcFace (2nd row), FaceNet (3rd row) and CurricularFace (4th row).

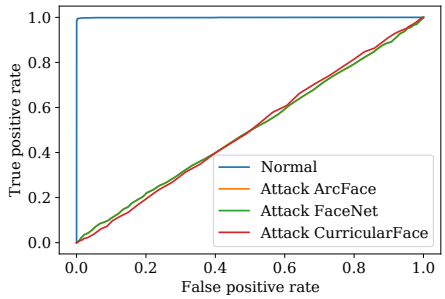


Figure 7: ROC curves on LFW benchmark after the FRS:es being attacked by FA attack.

Thresholds are set from 0 to 4 with a step size of 0.04 (400 thresholds in total); (3) For each threshold t, the distance d is calculated over all pairs. If $d \leq t$, the pair is a match; otherwise it is counted as a mismatch. (4) Using the counts of matches/mismatches and the labels of match/mismatch, the true positive rate (TPR), false positive rate (FPR), and the accuracy are calculated. In the evaluation for the FA attack in the LFW benchmark, we replace the facial images of pairs with mismatch labels by a selected image (in the pair) and its non-identical clone. With this replacement, it is still ensured that the visual looks of the mismatched pair are from different identifies but the distance d (of the pair) is pushed close to 0. As a result, most of the pairs in the mismatch category is classified as a match by the FRS:es. In Fig. 7, the effect of the FA attack brings the receiver operating characteristic (ROC) curve of all the FRS:es under attack close to diagonal, making the guess if a given pair is a match or not close to random (given the datasets 50-50 split).

Second, consider the change in identity. Fig. 8a shows the ID distance for pairwise FA attacks, MI attacks, and the random baseline for example experiments with CurricularFace. We note that the FA attack does not provide as good of a identity match as the MI attack but that the identity distances still typically are well below the identity recognition threshold of 1.25 (and the random baseline).

Third, to support that the identities we create have been moved away from the actual identities, we use the MS-SSIM distance to measure the semantic difference between images. Fig. 8b presents a comparison of both the FA and MI attack against the random baseline. We note that the FA attack (as desired) have much higher overlap with the baseline than the MI attack. We have observed similar results using SSIM (omitted). Fig. 8c presents the average values for all three metrics and scenarios. Again, the results clearly show that the FA attacks is able to create an identity with identity distance well below the 1.25 threshold while both of them achieving an SSIM and MS-SSIM distance close to the random baseline.

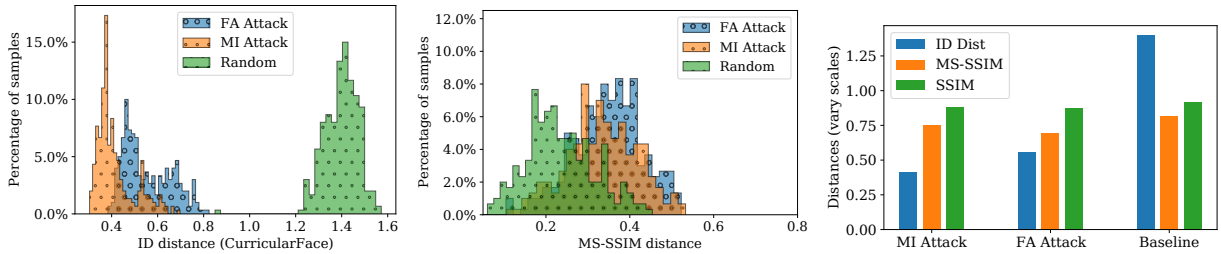
5.5 Comparing Face Embedding Models

Normalized identity reconstruction comparison: Thus far we have compared the recreated identities using the identity distances used by each model under attack. For fair comparison. we next use the Dlib [9] model (as a fourth model) to measure the identity distances of the reconstructed face from the ground truth faces. Fig. 9 shows these results. For completeness, we include both (a) distribution statistics, (b) average statistics, and (c) percentile plots. Here, it is important to note that a Dlib threshold (as evaluated in LWF) achieves an accuracy of 99.55% using a threshold of 0.6. In our case, ArcFace is able to achieve the lowest identity distances (including a narrower distribution, with most samples between 0.2 to 0.6). The distribution for FaceNet is similar. While both FaceNet and ArcFace have larger outliers, most faces extracted from these FRS would have distances below the identification threshold on Dlib. In comparison, CurricularFace is more spread out (e.g., main range 0.3–1.2) with higher average (0.4 vs 0.5).

Visual head-to-head comparison: We next use IdDecoder to visually compare different embedding models within the same framework. The basic methodology used here is to first reconstruct faces from the face embeddings of each system and then compare how much of the identity related and non-identity related information are directly or indirectly (e.g., via correlations with identity related aspects) included in the embedding.

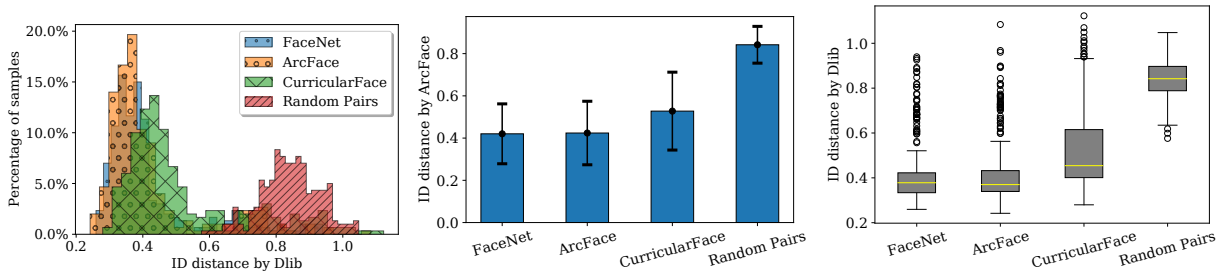
Fig. 10 presents the reconstructed faces of eight example identities (top row) for the three evaluated models: FaceNet (row two), ArcFace (row three), and CurricularFace (row four). In all the cases, as desired, the main identity features included in the inner facial area has been reconstructed relatively well. For example, for each face, we can recognize some degree of similarity between the reconstructed images and the ground truth images. (Again, in all cases this is achieved via a black-box attack using only the embeddings.)

We also see that the reconstructed images are mostly normalized in pose, lighting, and expression. This is an important aspect that helps understand the embeddings. For example, while face embeddings are trained using faces from the wild, with varying poses, lighting, expressions, etc., those non-identity-attribute-related aspects should ideally be excluded from the embedding. In contrast, attributes such as gender and skin color are preserved for most samples. We believe that these attributes typically are considered important attributes that are part of the identities themselves. Nonetheless, when we compare row by row, we can see light differences.



(a) Identity distance. (b) MS-SSIM distance. (c) Average distances.

Figure 8: Distance comparison of results using MI attack, FA attack, and the baseline of random pairs.



(a) Histogram of identity distances. (b) Average + standard deviations. (c) Box-plot of distances.

Figure 9: Cross comparison between face embedding models using Dlib to calculate pairwise distances.

Figure 10: Visual comparison between face embedding models. Comparing the ground truth (1st row) with the reconstructed faces using the MI attack on three models: FaceNet [1] (2nd row), ArcFace [2] (3rd row), and CurricularFace [3] (4th row).

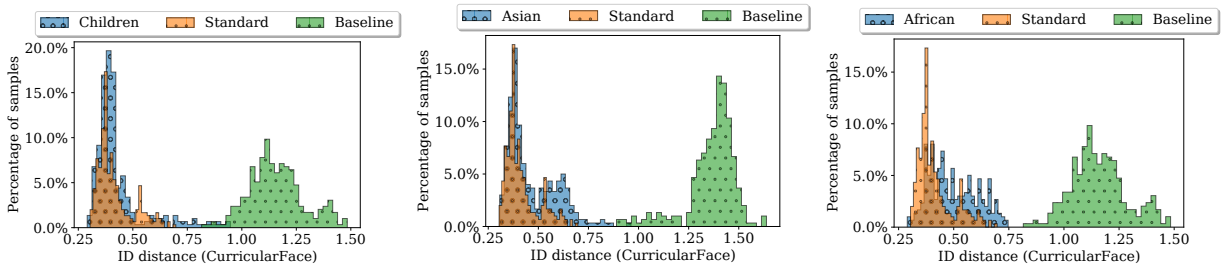
Comparison using minority groups: One weakness of many FRS:s is their ability to handle identities from minority groups. To gain some insights into how these biases (typically introduced due to lack of diversity in the training datasets) impact the ability of IdDecoder to reconstruct the original identities, we performed experiments using the UTK large-scale dataset [26]. Here, we focus on three minority groups, including young children of ages below 2, and two ethnicity groups: African and Asian. All three of these groups are often undetected or wrongly recognized by FRS:s [33].

Despite the biasness in the facial embedding, our IdDecoder still effectively reconstruct identities (in term of visual resemblance) for those minority groups. This can be observed by comparing the results for each of these three minority groups (shown in Fig. 11) with the corresponding example results from LFW dataset (see Fig. 5c) when using ArcFace. In most these cases, the reconstruction successfully capture the age correctly. However, when it comes to ethnicity we the biasness clearly visualized here. For examples, 3/4 Africans (Fig. 11c) appear to be Caucasian. Similarly, this happens for 1/4 of young children (Fig. 11a) and 1/4 of Asians (Fig. 11b).

Fig. 12 plots the relative identity distance (as calculated using CurricularFace) for the images associated with each minority group and compare this with the baselines of random faces and a standard set. While the minorities all have distance distributions slightly shifted toward the baseline, the majority of the distribution is overlapping with the standard one. This quantitative result might suggest that the FA attack may be more successful against minority

(a) Young children (b) Asian (c) Africans

Figure 11: Reconstruction examples (bottom row) using ArcFace embedding of sample images from minority groups (top row).



(a) Young children vs baselines. (b) Asian vs baselines. (c) African vs baselines.

Figure 12: Histogram comparisons of the identity distances observed when reconstructing images of (1) minority groups, (2) the standard dataset, and (3) the baseline of random pairs in the UTK dataset.

groups. However, it also shows that also the MI attack could be successful on the minority groups.

5.6 Comparison to other MI Attacks

We next compare the results achieved with IdDecoder to the results achieved by other MI attacks, each making different assumptions about the knowledge of the attacker. For the visual comparisons presented, we extracted images from the papers compare against and evaluated our results against the same images.

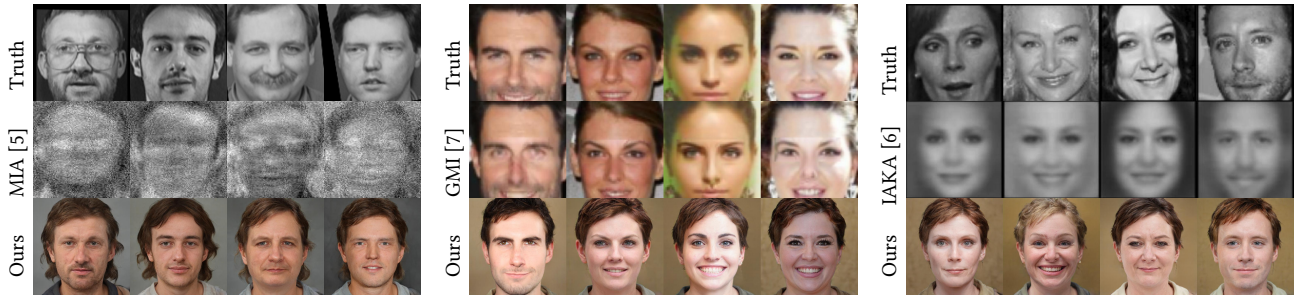
Original MI attack [5]: Fig. 13a compares our results inverting FaceNet's embedding to the original MI attack demonstrated against a FRS:s [5]. While Fredrikson et al. only considered classifier-based facial recognition models, they presented several such models, including stacked denoising autoencoder (DAE), multilayer perception, and softmax regression. Although they are all shallow models and are not comparable to the state-of-the-art, we choose to include results for the DAE version in a black-box setting since the threat model is the most advanced one and assumes an attack setting closer to ours than the others. As seen in the figure, the reconstructed version from their attack (called MIA here) are almost unrecognizable, while our results are more clearly observed, although the low resolution and black-and-white images were found to present challenges also to our model (as our training used Celeb-HQ).

White-box attack comparison: We next compare our results against the work by Yang et al. [6] and Zhang et al. [7]. These results are shown in Fig. 13b and Fig. 13c. Like us, Zhang et al. [7] use a GAN. However, they assume a white-box setting and require hard-to-obtain auxiliary information such as a blurry or masked version of the target face. This makes their attack much less practical than ours. In a sense, their method is similar to image in-painting, where parts of the images are covered/blurred and the task is to reconstruct the covered/blurred version. In addition to a much more relaxed requirement and a totally different attack scenario than ours, we have found that the visual results (e.g., those presented by the authors) can look really good while other results are inconsistent and distorted. We expect this to be due to the difficult task of training GANs. Yang et al. [6] present another white-box attack using auxiliary information. A comparison with this work is presented in Fig. 13c. Compared to our results, their results are blurrier and often resembles an average face, while our approach provides somewhat clearer representation of the individual faces.

Table 1: Quantitative Comparison of our method with MIA [5] and IAKA [6].

Method	ID (w. Dlib)	MS-SSIM Dist	LPIPS
MIA [5]	1.04 ± 0.11	0.61 ± 0.22	0.698 ± 0.32
IAKA [6]	0.45 ± 0.32	0.32 ± 0.01	0.55 ± 0.15
Ours	**0.30 ± 0.04**	**0.27 ± 0.019**	**0.28 ± 0.02**

Numerical comparison with open-source solutions: Table 1 presents a numerical comparison with the two related works that provide open-source codes at the time of writing. Before discussing these results a few important notes are needed. First, to represent the work by Fredrikson et al. [5] we used their white-box setting with DAE model. While this model could not be used in practice for the complex task as facial recognition, it was the only model that could provide reasonable results, as the other versions (e.g., using soft-max regression and MPL) do not result in recognizable faces (e.g., as seen in Fig. 13a). Second, we use the relaxed version for the purpose of lower-bound evaluation. Yet, we outperform both this white-box attack and the white-box attack by Yang et al. [6] (Adversarial Neural Network Inversion via Auxiliary Knowledge Alignment or IAKA for short) with regards to all three performance metrics: ID distance (calculated using Dlib [9]), and a distance-based version of MS-SSIM (i.e., one minus the similarity metric) [28], and LPIPS [20]. Regardless of which distance metric is used, we achieve the smallest distances. Most importantly, we achieve a identity distance well below Dlib's threshold of 0.6 (with margin as seen by small standard deviation).



(a) Comparison with the original MI attack [5] using a DAE model in black-box setting. **(b) Comparison with GMI [7] using classifier model + auxiliary information + white-box.** **(c) Comparison with IAKA [6] in black-box setting.**

Figure 13: Comparison with related work using ArcFace model in black-box setting. We show ground truth (1st row), their results (2nd row), and our results (3rd row).

Comparison of high-level properties: Finally, Table 2 summarizes the main differences between our framework (i.e., IdDecoder when used for MI attack) and the related works that perform MI attacks. Three main differences are highlighted here: (1) the attack settings, i.e., white-box vs. black-box, (2) whether the attack requires auxiliary information such as a blurred/covered image of the target victim, and (3) whether the attack is proposed to be applied against state-of-the-art (mostly face embedding models) or only classifier models. We note that our model is a black box attack, does not require any auxiliary information, and can be applied on state-of-the-art embedding models.

Table 2: Comparison to the related works

Method	Settings	Auxiliary	Attack SOTA
Our	Black-box	No	Yes
MIA [5]	Black/white-box	No	No
IAKA [6]	White-box	Yes	No
GMI [7]	White-box	Yes	No
Cole et al. [8]	White-box	No	Yes

6 RELATED WORK

Inverting face embeddings: Early works inverting face embeddings proposed the use of feed-forward networks or random searches to invert features from the CNNs [34] and the embeddings [35, 36]. Although successful at some level, the results produced by most of these early attacks lack details and, in most cases, result in unrecognizable faces. The use of GANs [37, 38] have been shown to improve the details of the reconstructed faces but these works still result in some blurriness and are in some cases un-usable. Perhaps the most comparable results to ours are the works by Vendrow et al. [39] and Cole et al. [8]. However, the optimization searching algorithm of Vendrow et al. [39] cannot capture the neutralization features of face embedding and the method by Cole et al. [8] requires access to the models where the embeddings are not used directly but the features in earlier layers before the final output are used (white-box attack). In contrast, our method reconstructs realistic, normalized faces without any access or prior knowledge about the underlying model.

Model Inversion Attack: First introduced by Fredrikson et al. in 2014 [14], Model Inversion (MI) attacks were originally proposed against a linear regression model to recover sensitive features of genetic markers. The attack received more attention when it was demonstrated against a FRS based on a shallow neural network [5]. The attack exploits confidence values revealed along with predictions made by the attacked model. While the attack has higher success rate than random guessing, the attack only works for a multiple-class classifier model and the results in most cases are barely recognizable as the original identity.

More recently, a new variant of the attack [7] based on GANs were demonstrated to work against more complex facial recognition based on deep neural networks (DNNs). However, this attack is performed in a white-box setting, in which the attacker requires additional knowledge such as a blurred or masked copy of the target image. This significantly reduces the practicality of the attack. Another distinguishing difference compared to our attack is that the FRS they consider is based on a multiple-classes classifier that does not take advantage of the newly proposed facial cognition system that extracts and use face embeddings.

Generative Adversarial Networks (GANs): We have recently seen big breakthroughs in GANs [15]. After learning the original distribution of a training dataset with a high degree of generativity, these models can often be used to generate new realistic samples that are almost indistinguishable from the real data. GANs have been used for generating complex datasets, including text [40, 41] and music [42, 43]. Perhaps the most impressive results of GANs have been their ability to synthesize and generate realistic images by StarGAN [44], PGGAN [45], and StyleGAN [16–18]. These models are capable of generate realistic images of complicated domain such as cars, human faces, and animals. Besides a high-quality image generator, StyleGAN brings another benefit: its highly disentangled latent space. The latent space of StyleGAN can be seen as a set of feature layers in the middle of its architecture. The highly disentangled nature of this space has allowed researchers to make small changes in the latent space that only effect a certain feature of a face (e.g., the hair color, face expression, age, etc.) [46–48] or anonymizing the facial identity without noticeably changing the rest of the image [49].

7 CONCLUSIONS

In this paper, we first presented the IdDecoder framework, capable of effectively synthesizing realistic, neutralized face images from face embeddings. The framework incorporates both the use of a mapper network that combines up-scaling and several mappers responsible for different layers in the latent space as well as an

optimizer that builds on some of the insights obtained from the design of the mappers. Central to the success are also the use of initialization vectors that help normalize the faces. These play a similar role as the strict requirements often associated with passport photos (e.g., with regards to pose, background, etc.). Second, using IdDecoder, we presented a black-box model inversion attack that allows the attacker to reconstruct a realistic face image that is both visually and numerically (as determined by the FRS:s) recognized as the same identity as the original face used to create a given face embedding. Given that these systems typically do not conceal the embeddings and some even provide APIs to obtain them in clear text, this attack raises significant privacy concerns regarding the protection of both the embeddings of the gallery set and the queries made within systems using these FRS:s, for example. Third, using a relaxed version of the loss function used to perform the model inversion attack, we presented a false acceptance attack in which we train IdDecoder to create the face of an alternative identity that is visually different than the original identity but that has a similar embedding, and that the FRS would recognize as being of the same identity. This attack raises significant concerns as it can be used to reduce the security and reputation of deployed FRS:s. Overall, in addition to the privacy and security concerns raised by our demonstrated attacks, the high efficiency achieved by the attacks (e.g., compared to prior work) raises new questions about when and how to best protect the embeddings and the integrity of the FRS:s in general. Our code can be found here: https://github.com/minha12/IdDecoder

ACKNOWLEDGMENTS

This work was supported by the Swedish Research Council (VR) and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE CVPR*, 2015.
[2] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF CVPR*, 2019.
[3] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang, "Curricularface: adaptive curriculum learning loss for deep face recognition," in *Proc. CVPR*, 2020.
[4] Clarifai, "Ai face detection model - clarifai," 2022.
[5] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. ACM CCS*, 2015.
[6] Z. Yang, E.-C. Chang, and Z. Liang, "Adversarial neural network inversion via auxiliary knowledge alignment," *arXiv preprint arXiv:1902.08552*, 2019.
[7] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proc. CVPR*, 2020.
[8] F. Cole, D. Belanger, D. Krishnan, A. Sarna, I. Mosseri, and W. T. Freeman, "Synthesizing normalized faces from facial identity features," in *Proc. CVPR*, 2017.
[9] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
[10] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE CVPR*, 2014.
[11] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. BMVC*, British Machine Vision Association, 2015.
[12] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *Proc. IEEE FG*, 2018.
[13] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," in *Workshop on faces in'Real-Life'Images at ECCV*, 2008.
[14] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proc. USENIX Security Symposium*, 2014.
[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
[16] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF CVPR*, 2019.
[17] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proc. IEEE/CVF CVPR*, 2020.
[18] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," in *Proc. NeurIPS*, 2021.
[19] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," in *Proc. IEEE/CVF CVPR*, 2021.
[20] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. CVPR*, 2018.
[21] S. Guan, Y. Tai, B. Ni, F. Zhu, F. Huang, and X. Yang, "Collaborative learning for faster stylegan embedding," *arXiv preprint arXiv:2007.01758*, 2020.
[22] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. ECCV*, 2016.
[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
[24] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *Proc. IEEE CVPR*, 2020.
[25] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. ICCV*, 2015.
[26] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proc. IEEE CVPR*, 2017.
[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
[28] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. ACSSC*, 2003.
[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012.
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016.
[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[32] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," *Dept. Comp. Sci., UMass, MA, Tech. Rep*, 2014.
[33] I. D. Raji, T. Gebru, M. Mitchell, J. Buolamwini, J. Lee, and E. Denton, "Saving face: Investigating the ethical concerns of facial recognition auditing," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 145–151, 2020.
[34] A. Dosovitskiy and T. Brox, "Inverting visual representations with convolutional networks," in *Proc. IEEE CVPR*, 2016.
[35] A. Zhmoginov and M. Sandler, "Inverting face embeddings with convolutional neural networks," *arXiv preprint arXiv:1606.04189*, 2016.
[36] A. Razzhigaev, K. Kireev, E. Kaziakhmedov, N. Tursynbek, and A. Petiushko, "Black-box face recovery from identity features," in *Proc. ECCV*, 2020.
[37] Z. Li and Y. Luo, "Generate identity-preserving faces by generative adversarial networks," *arXiv preprint arXiv:1706.03227*, 2017.
[38] C. N. Duong, T.-D. Truong, K. Luu, K. G. Quach, H. Bui, and K. Roy, "Vec2face: Unveil human faces from their blackbox features in face recognition," in *Proc. IEEE/CVF CVPR*, 2020.
[39] E. Vendrow and J. Vendrow, "Realistic face reconstruction from deep embeddings," in *NeurIPS 2021 Workshop Privacy in Machine Learning*, 2021.
[40] W. Fedus, I. Goodfellow, and A. M. Dai, "Maskgan: better text generation via filling in the_," *arXiv preprint arXiv:1801.07736*, 2018.
[41] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proc. AAAI*, 2017.
[42] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.
[43] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang, "Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment," in *Proc. AAAI*, 2018.
[44] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE/CVF CVPR*, 2018.
[45] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.
[46] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "Ganspace: Discovering interpretable gan controls," *arXiv preprint arXiv:2004.02546*, 2020.
[47] Z. Wu, D. Lischinski, and E. Shechtman, "Stylespace analysis: Disentangled controls for stylegan image generation," in *Proc. IEEE/CVF CVPR*, 2021.
[48] Y. Shen, C. Yang, X. Tang, and B. Zhou, "Interfacegan: Interpreting the disentangled face representation learned by gans," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2020.
[49] M.-H. Le and N. Carlsson, "Styleid: Identity disentanglement for anonymizing faces," *Proceedings on Privacy Enhancing Technologies*, vol. 1, 2023.