# The Many-faced God: Attacking Face Verification System with Embedding and Image Recovery

Mingtian Tan, Zhe Zhou*
{18210240176,zhouzhe}@fudan.edu.cn
Fudan University

Zhou Li
zhou.li@uci.edu
University of California, Irvine

## ABSTRACT

Face verification system (FVS), which can automatically verify a person's identity, has been increasingly deployed in the real-world settings. Key to its success is the inclusion of face embedding, a technique that can detect similar photos of the same person by deep neural networks.

We found the score displayed together with the verification result can be utilized by an adversary to "fabricate" a face to pass FVS. Specifically, embeddings can be reversed at high accuracy with the scores. The adversary can further learn the appearance of the victim using a new machine-learning technique developed by us, which we call embedding-reverse GAN. The attack is quite effective in embedding and image recovery. With 2 queries to a FVS, the adversary can bypass the FVS at 40% success rate. When the query number raises to 20, FVS can be bypassed almost every time. The reconstructed face image is also similar to victim's.

**ACM Reference Format:**
Mingtian Tan, Zhe Zhou and Zhou Li. 2021. The Many-faced God: Attacking Face Verification System with Embedding and Image Recovery. In *Annual Computer Security Applications Conference (ACSAC '21), December 6–10, 2021, Virtual Event, USA*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3485832.3485840

## 1 INTRODUCTION

Machine learning (ML), especially Deep Learning based on Deep Neural Networks (DNN), has transformed many important application domains, like computer vision, language processing and speech recognition. In certain tasks, DNN can achieve far better result comparing to the human expert, thanks to its capability of modeling the complex relation between input and output domains. Apart from high accuracy, ML is easy to implement, which also contributed to its popularity. Usually a deep learning model costs developers only several hundreds of Python codes but can already produce satisfied accuracy.

Face verification is such an application scenario supported by ML. State-of-the-art face embedding schemes like Facenet can achieve over 99% accuracy. Motivated by such result, face verification systems (FVS) powered by face embedding are widely deployed at places like border control [8, 17, 44, 47], company entrance [2, 29] and mobile device [4, 50]. Its success is mainly attributed to its convenient workflow: after the user enrolls in FVS with her ID and face images, next time, the same user can be quickly verified based on the embedding.

Unfortunately, our study revealed a severe confidentiality issue of the deployed FVS. By carefully probing a targeted FVS with a set of faces and observing the responses, not only an attacker can "create" a face that successfully passes the check of FVS, she can also recover the face image of a victim enrollee. Our attack does not exploit any specific bug of FVS system nor require access to enrollee's image (in fact, the face image is never stored by FVS). In particular, our attack is based on three unique insights about FVS and face embedding: 1) No matter what the verification result is, information about the victim enrollee leaks to the attacker. 2) Such information can be accumulated so that the face embedding, the internal representation created by the embedding model about a user, can be recovered. 3) The face embedding is highly sensitive, because an attacker can reconstruct the input image with high fidelity under its guidance. Below we elaborate the three insights.

**1) Information leakage from FVS.** For some FVS, every time the verification result ("pass" or "fail") is displayed to an attested person, the score is also displayed for the debugging purposes, reflecting how far/close the person's image to the enrollee's. The score is directly related to the distance on the embedding space.

Every time the similarity to attested person is displayed, the system leaks a small portion of information about the claimed user's face. The similarity could help attackers to recover the embedding (a vector representing a face) of the profile photo once enough information is collected.

**2) Embedding recovery from leakage.** At the first glance, reversing the victim enrollee's face from FVS score seems infeasible, as the information contained by it is negligible. However, the information can be accumulated, when the attacker probes FVS with different images. One of our key findings is that *when the number of inquiry images equals to the dimensions of the embedding model, the victim's embedding can be recovered without error*, mainly because an embedding is a high-dimension vector which still obeys algebraic geometry theorems. By formulating the embedding distances with equations on Euclidean space, the root of equations corresponds to the exact embedding. Furthermore, we found through a dimension-reduction approach based on PCA (Principal Component Analysis), the adversary can issue much less queries to recover a similar embedding.

**3) Image recovery with embedding.** With the embedding, the attacker is supposed to reconstruct the victim's photo. However, face embedding is a complex, non-linear and lossy mapping from an input sample. Reversing such mapping is quite challenging,

*Zhe Zhou is the Corresponding Author.


*ACSAC '21, December 6–10, 2021, Virtual Event, USA*

ACM ISBN 978-1-4503-8579-4/21/12…$15.00
https://doi.org/10.1145/3485832.3485840

which has not been resolved by prior works. We propose a novel approach based on generative adversarial network (GAN) for this task. Classical GAN models reconstruct images from noise input or a pre-defined label but none of them deal with unseen input. Therefore, we design a new embedding-reverse GAN (or erGAN) with the generator and loss function tailored to the embedding input.

**Major results.** We evaluate the embedding recovery (named **EmbRev**) and face recovery (named **ImgRev**) modules. The overall result proves learning faces enrolled in an FVS just from scores is feasible and the attack is practical.

For **EmbRev**, we evaluated over 13,000 images contained by the LFW dataset on 4 different embedding models. When the query number equals to the embedding dimensions (*e.g.*, 128 queries when attacking Facenet-128), the embedding can be recovered nearly perfectly (the error margin is due to floating-point precision). When reducing the query number to half, the error distance is still small, at only 0.1 in average, which is far smaller than the distance threshold of the targeted models (*e.g.*, 1.28 for Facenet-128). In fact, **only 2 queries are sufficient to help the adversary byass FVS at 40% chances under whitebox setting**. We also found **EmbRev** achieves consistent performance when the image is distorted or some digits of the FVS score are hidden.

For **ImgRev**, we evaluated with the images from CelebA dataset. Our result shows the images reversed from the perfect embeddings can pass all 4 evaluated embedding models with over 90% success rate. Furthermore, based on FID metrics, the quality of our recovered images are considered quite satisfactory (*e.g.*, 34 for Clarifai-1024), considering that adding a pair of eye glasses easily raises FID over 47 [63]. When the recovered embedding contains errors, *e.g.*, due to the reduced query number, the result maintains the same level. The consequence of **ImgRev** is severe. Take a face verification based door entrance system as an example, an attacker can claim to be an arbitrary enrollee (victim) and pass the entrance with the recovered photo. Moreover, **ImgRev** eventually can help attackers infer similar photos to all enrollees' photos stored in the FVS database, leading to outstanding privacy leakage.

We have reported our discoveries to stake-holders like Clarifai. The code of this project will be released at a GitHub repo[1].

We summarize our contributions as follow:

- We identified that the confidentiality of FVS enrollees is under threat when the adversary probes the FVS with different images.
- We presented a new attack against face embedding. Our attack is able to recover a sensitive face embedding with only a few to dozens of queries.
- We developed a new DNN model based on GAN, which is able to reconstruct an image close to victim's from a recovered embedding.
- We evaluated our attack with state-of-the-art embedding models and real-world face dataset.


**(a) A self-service FVS in Chinese Entry & Exit Bureau.** **(b) A FVS app [14].**

**Figure 1: Two examples of FVS. To notice, the similarity scores are displayed.**

## 2 BACKGROUND

### 2.1 Face Verification

A face verification system (FVS) takes a digital image or video through camera as input and matches it with the database of face images to verify the claimed identity. It has been widely deployed by government for surveillance and border control [8, 17, 44, 47], enterprise for attendance tracking [2, 29] and mobile device for owner authentication [4, 50]. When the verification process is initiated, the face detection module discovers the face region and sends it to the face matching module, which computes a score between the captured face image and the enrolled face images to decide whether the person can be authenticated.

However, as previous work identified [37], face verification is vulnerable under media-based facial forgery (MFF) attack, where the adversary captures the victim's face (e.g., from social network) ahead and replays the crafted photo/video. To detect such forged face, *liveness detection system* was proposed. It either uses sensors (e.g., accelerometer and gyroscope) or challenge-response protocol (e.g., asking the user to smile) to assign a liveness score about the inputted image/video [34, 36, 55, 57]. Yet, its effectiveness is questionable when the adversary can wear a mask with the victim's face printed [16]. In this work, we focus on bypassing FVS with static image. To bypass live detection system, the methods proposed previously [16] can be leveraged.

**Face embedding.** The accuracy of face verification highly depends on the face matching module. Specifically, it should give high similarity score to the face images of the same person but low score to those of different persons. Nowadays, face embedding models like Clarifai [10] (online service) and Facenet (open-source implementation) [3, 51] are integrated to build the face matching module. Face embedding is a Deep Convolutional Neural Network trained with face images collected from a pool of participants (each participant can have multiple images). Given an image, the face embedding model will map it to a vector of $N$ dimensions (*e.g.*, 128 or 512 for Facenet [51] and 1024 for Clarifai [10]), which is also called *embedding*. The deployed FVS usually uses pre-trained model (*e.g.*, trained with public face dataset like CASIA-WEBFACE[68]). In enrollment stage, FVS stores the embedding and its user ID (*e.g.*, employee)

[1] https://github.com/BennyTMT/DL_Privacy

into the biometric database, which is kept as secret. In verification stage, the embedding of the attested person will be compared to the embedding of his enrolled profile photo under the provided ID. The similarity is typically computed using L2 distance or Cosine distance and the person is authenticated if the similarity is over a threshold. In addition to face verification, face embedding has also been leveraged to find similar persons and techniques like Locality Sensitive Hashing (LSH) [21] can be leveraged.

## 2.2 Adversary Model

The primary goal of our adversary is to impersonate another person who has enrolled in a deployed FVS and bypass the check. Specifically, the adversary intends to forge a face image with minimum distance to victim *on the embedding plane*. Such attack can deal great damage to public safety. For example, an enlisted terrorist can escape into country's border which deploys self-service FVS (shown in Figure. 1a). The secondary goal of the adversary is to learn the appearance of a victim without her consent, which violates her privacy. In other words, the forged image should also look realistic, with minimum distance to the victim *on the image plane*. Adversarial examples do not meet this goal as they do not need to look similar to victims but attackers.

Our attack consists of five steps. (1) We assume victim's ID has been obtained, *e.g.*, through searching public ID database. The adversary comes to the FVS, enters the ID of the targeted victim, and initiates the face verification process. (2) The verification result (it should be "fail") and score are displayed, which leaks information about the victim. To gain more information, multiple scores according to different attempts are collected, which can be done by showing different face images or recruiting a group of people to approach the FVS. (3) The adversary reconstructs the embedding of the victim with the tested faces and their scores. (4) Victim's face image is recovered through a generative model. (5) The adversary prints out the generated face image (e.g., as a mask) and wears it to bypass FVS. Figure 2 illustrates our attack process.

**Leakage of FVS score.** The calculated distance, from some FVS developers' perspective, is not sensitive. An example we encountered is a self-service machine that was deployed at the *Chinese entry and exit bureau* (the counterpart of immigration or boarder inspection of some countries, and also part of the police system). This machine authenticates users with their faces before other tasks. The machine directly displays the similarity on the screen (see Figure 1a). Another example is an app that directly shows the matching score on its UI to users, as shown by Figure 1b. No matter if similarity, score or confidence level displayed, they are eventually variants of embedding distance, through which attackers can infer the distance.

**White-box adversary.** In this scenario, the adversary knows the structure of the face embedding model $f$ used by the targeted FVS, including layers, hyper-parameters and weights. The adversary can conduct the attack with the help of $f$.

While this assumption seems strong, meeting such requirement is feasible in many cases. For instance, the adversary can purchase or download the same FVS system and reverse engineer the model structure. In addition, open-source face recognition library like Open face [3] has been used by many FVS and attacking such FVS is even easier as the model can be directly extracted without reverse-engineering.

To notice, white-box adversary is also covered by prior works about machine-learning confidentiality [18, 19, 58] and the assumption is similar.

**Black-box adversary.** When the FVS is close-source or its open-source implementation is not available, the adversary will not be able to directly replicate $f$. We consider one situation that the adversary is able to access the embedding produced by $f$ without knowing its structure. Due to the advent of Cloud-based Machine Learning as a Service (MLaaS), there have been many FVS using APIs of an online embedding service for face verification. One famous example is Clarifai [10], which has pre-trained models with very high accuracy and sells its access (*i.e.*, returning the face embedding vector given an inquiry image) to customers [11]. When the adversary identifies the MLaaS model used by the targeted FVS (*e.g.*, through sniffing its network traffic and identifying the destination IP address), she can query its API with forged images to obtain the embeddings outputted by $f$, in addition to the displayed score.

**No-box adversary.** In the worst case, the adversary cannot obtain the access to the implementation of the targeted FVS or even its MLaaS API, which we call "no-box" adversary. Learning $f$ or the embeddings becomes impossible. However, as we will later show, by attacking another embedding model, the adversary is still able to recover victm's embedding.

## 2.3 GAN

The core step of our attack is to reconstruct the victim's face image from the recovered embedding, which can be categorized as a generative task (in contrast to prediction). We leverage Generative Adversarial Network (GAN) [22] to fulfill this task, which has shown great successes in synthesizing plausible image [22], sound [40] and text [72].

GAN consists of two neural networks: a generator and a discriminator. The generator maps randomized input sampled from a pre-defined latent space (or "noise") to a data distribution of interest in the target space. The discriminator determines if a data distribution is authentic or synthesized by the generator. The training goal of the generator is to increase the error rate of (or "fool") the discriminator, while the goal of the discriminator is to maintain high accuracy in distinguishing the data distributions. The generator and the discriminator are trained in turn to minimize the outcome of a loss function, *e.g.*, minimax loss [22] or Wasserstein loss [5].

There are also a bunch of famous GAN variant works, like image to image translation [30], image to image translation without paired data [75].

# 3 EMBEDDING RECOVERY

In this section, we describe **EmbRev**, the module developed by us to infer the face embedding of a victim based on the score displayed by FVS. To summarize, **EmbRev** can recover the *exact* embedding vector (*e.g.*, 128 dimensions under Facenet-128) when a relatively large set of scores has been obtained (*i.e.*, the same number as the embedding dimension) through "querying" FVS. When the number

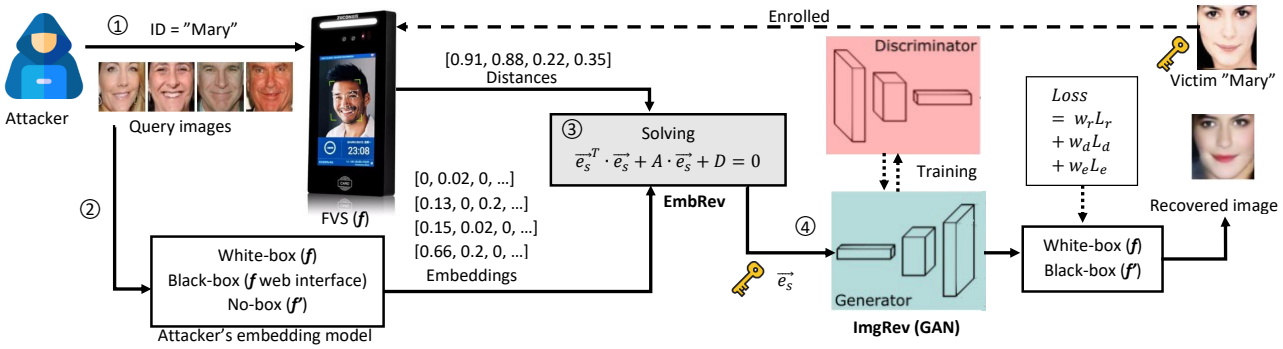


**Figure 2: Overview of our attack. 1) The attacker queries FVS with a set of face images to obtain the FVS scores, which are converted to embedding distance. 2) The attacker inputs the images to the same embedding model as FVS ($f$) or a surrogate model $f'$ to obtain their embeddings. 3) The distances and embeddings are inputted into EmbRev to recover victim's face embedding $\vec{e_s}$. $A$ and $D$ are generated from the distances and embeddings. 4) The sensitive embedding $\vec{e_s}$ is inputted to the generator of ImgRev to recover victim's face.**

of queries is limited, **EmbRev** is still able to approximate victim's embedding.

## 3.1 EmbRev with Equation-Solving

We denote the sensitive embedding as $\vec{e_s}$, which is generated by the embedding model $f$ of FVS, *i.e.*, $\vec{e_s} = f(x_s)$, where $x_s$ is victim's image. $n$ is the dimension of embedding (*e.g.*, 128). We assume the adversary has issued $m$ images $(x_1, x_2, ..., x_m)$ to FVS and obtained a series of scores $s_1, s_2, ..., s_m$, which can be converted to distances $d_1, d_2, ...d_m$ ($s_i + d_i = 1$ for the simplest case). In the meantime, the adversary also converts the query images to embeddings, denoted as $\vec{e_1}, \vec{e_2}, ..., \vec{e_m}$, in order to learn $\vec{e_s}$.

When the adversary knows the embedding model $f$ of FVS (white-box adversary), $\{\vec{e_1}, \vec{e_2}, ..., \vec{e_m}\}$ can be easily constructed with $f(x_1), f(x_2), ..., f(x_m)$. When the adversary only has access to the MLaaS API of $f$ (black-box adversary), $\{\vec{e_1}, \vec{e_2}, ..., \vec{e_m}\}$ can be learnt as well by reading the API response. In section 3.3, we discuss the no-box adversary.

*Our first finding is that $\vec{e_s}$ can be fully recovered when $m = n$.* When the $n = 2$, the proof is straightforward. In this case, $\vec{e_s}$, $\vec{e_1}$ and $\vec{e_2}$ can be considered as points in two-dimensional Euclidean space, and $\vec{e_s}$ must be on the intersection of the two circles extended from $\vec{e_1}$ and $\vec{e_2}$ (with radius $d_1$ and $d_2$). The intersection can have one or two points. Finding the intersection is actually the same as solving the equations of $||\vec{e_s} - \vec{e_1}|| = d_1$ and $||\vec{e_s} - \vec{e_2}|| = d_2$ where $\vec{e_s}$ is the unknown variable. When $n > 2$, learning the root of $e_s$ becomes non-trivial as $n$ equations will be involved, as shown in Equation Set 1.

$$\begin{aligned} ||\vec{e_s} - \vec{e_1}|| &= d_1 \\ ||\vec{e_s} - \vec{e_2}|| &= d_2 \\ &... \\ ||\vec{e_s} - \vec{e_n}|| &= d_n \end{aligned} \tag{1}$$

Through careful analysis, we found Equation Set 1 is still solvable. When L2 distance[2] is used, we can convert Equation Set 1 to Equation 2 below after squaring each equation, assuming $\vec{e_s}$, $\vec{e_1}$, ..., $\vec{e_n}$ are column vectors.

$$\vec{e_s}^\intercal \cdot \vec{e_s} + A \cdot \vec{e_s} + D = 0 \tag{2}$$

where $A = -2 \cdot \{\vec{e_1}, \vec{e_2}, ..., \vec{e_n}\}^\intercal$ and $D = \{\vec{e_1}^\intercal \cdot \vec{e_1} - d_1^2, \vec{e_2}^\intercal \cdot \vec{e_2} - d_2^2, ..., \vec{e_n}^\intercal \cdot \vec{e_n} - d_n^2\}^\intercal$.

**Euclidean distance.** To solve Equation 2, we firstly introduce a new scalar variable $z$ and assign it with $\vec{e_s}^\intercal \cdot \vec{e_s}$, where $\vec{e_s}$ is a column vector. With the introduction of $z$, Equation 2 can be converted into Equation 3 and Equation 4 in which the right-hand side has no $\vec{e_s}$.

$$z + A \cdot \vec{e_s} + D = 0 \tag{3}$$

$$\vec{e_s} = -A^{-1} \cdot (D + z) \tag{4}$$

Now we replace $\vec{e_s}$ in $z = \vec{e_s}^\intercal \cdot \vec{e_s}$ with Equation 4, so Equation 6 can be derived.

$$\begin{aligned} z &= \vec{e_s}^\intercal \cdot \vec{e_s} && (5) \\ &= (D + z)^\intercal (A^{-1})^\intercal \cdot A^{-1} \cdot (D + z) \\ &= D^\intercal B D + z \cdot \vec{1}^\intercal B D + z \cdot D^\intercal B \cdot \vec{1} + z^2 \cdot \vec{1}^\intercal B \cdot \vec{1} && (6) \end{aligned}$$

where $B = (A^{-1})^\intercal \cdot A^{-1}$ and $\vec{1} = \{1, 1, ..., 1\}^\intercal$ with $n$ 1's.

Because $z$ is a scalar value (it equals to the multiplication of a row vector and a column vector), Equation 6 is a quadratic function with $z$ as the unknown variable. Therefore, $z$ has up to two roots, as shown by Equation 7. For $\vec{e_s}$, up to two roots are available as well because of Equation 4. The roots are shown in Equation 8, by assigning Equation 7 into $z$ of Equation 4.

$$z = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{7}$$

[2] $d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$, where $p$ and $q$ are two vectors of $n$ elements.

where $a = \vec{1}^\intercal B \cdot \vec{1}$, $b = \vec{1}^\intercal BD + \cdot D^\intercal B \cdot \vec{1} - 1$, $c = D^\intercal BD$, and $B = (A^{-1})^\intercal \cdot A^{-1}$.

$$\vec{e_s} = -A^{-1} \cdot (D + \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}) \quad (8)$$

where $a = \vec{1}^\intercal B \cdot \vec{1}$, $b = \vec{1}^\intercal BD + \cdot D^\intercal B \cdot \vec{1} - 1$, $c = D^\intercal BD$, $B = (A^{-1})^\intercal \cdot A^{-1}$ and $\vec{1} = \{1, 1, ..., 1\}^\intercal$ with $n$ 1's.

When $b^2 = 4ac$, $e_s$ is very likely to have only one meaningful root. We used Matlab to test **EmbRev**, and did not find the case of two meaningful roots after trying many instances. When there are two real roots, the incorrect one fall out of the normal distribution of embeddings, *i.e.*, has large norm. Therefore, $e_s$ can be uniquely identified.

**Cosine Distance.** For embedding schemes choosing Cosine distance[3], $\vec{e_s}$ can be inferred as well by solving the equation below.

$$A \cdot \frac{\vec{e_s}}{|\vec{e_s}|} = D \quad (9)$$

where $A = \{\frac{\vec{e_1}}{|\vec{e_1}|}, \frac{\vec{e_2}}{|\vec{e_2}|}, ..., \frac{\vec{e_n}}{|\vec{e_n}|}\}^\intercal$ and $D = \{1 - d_1, 1 - d_2, ..., 1 - d_n\}^\intercal$.

There is only one root for $\vec{e_s}$, which is $A^{-1} \cdot D \cdot |\vec{e_s}|$. Though $|\vec{e_s}|$ cannot be derived, different value has no impact to the final result, as the $|\vec{e_s}|$ will be normalized by the generator of **ImgRev**. Therefore, we set $|\vec{e_s}| = 1$.

Overall, our result shows face embedding cannot be secured when the adversary can query the FVS with a set of images and record all the returned scores. Essentially, face embedding "compresses" an image to a vector in a much smaller latent space (*e.g.*, 128 or 512 dimensions for Facenet). The mapping is deterministic and the entropy is significantly reduced, as such the embedding is much easier to recover than its source image.

## 3.2 Reducing Query Number

Though effective, running **EmbRev** can be costly as $n$ queries are required. Under certain scenarios like self-service FVS at border, obtaining hundreds of distances might be impossible for the adversary. On the other hand, *we found reducing the dimension of embedding does not have big impact on the embedding model.* Therefore, the adversary can reconstruct an embedding with smaller dimensions but still pass face verification.

**Impact of embedding dimension.** Firstly, we carefully reviewed the Facenet embedding scheme [52]. It turns out when increasing the dimension from 64 to 128, under L2 distance, Facenet only gains 1 percent higher accuracy (86.8% vs 87.9%, shown in Table 5 of [52]). Interestingly, when the dimension is increased to 256 and 512, the accuracy degrades (87.7% and 85.6%). The result indicates small dimension volume like 64 can accommodate most of the key information of a face image. In fact, one possible explanation about the accuracy plateau or decline is the use of dropout [56] when training the embedding models. To avoid over-fitting, developers intentionally shut off some neurons during a training iteration, which pushes different neurons to generate similar output and introduces high information redundancy to a layer's output.

[3] $d(p, q) = 1 - \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\sqrt{\sum_{i=1}^{n} q_i^2}}$, where $p$ and $q$ are two vectors of $n$ elements.

To further understand how information is stored in the embedding, we generate Facenet-128 embeddings for 400 randomly selected images from the LFW (Labelled Faces in the Wild) dataset [35] and then use Singular Value Decomposition (SVD) to extract the key components of each embedding. SVD is a variant of Principal Component Analysis (PCA) over matrices, which transforms possibly correlated data into linearly uncorrelated variables. With SVD, a $m$-by-$n$ matrix $M$ can be decomposed to the product of three matrices, *i.e.*, $M = U \cdot \Sigma \cdot V^\intercal$, in which $U$ and $V^\intercal$ are unitary matrices and $\Sigma$ is a rectangular diagonal matrix. By replacing $\Sigma$ with $\tilde{\Sigma}$ which has $r$ largest singular values, we can approximate $M$ with another $r$-rank matrix $\tilde{M} = U \cdot \tilde{\Sigma} \cdot V^\intercal$. In our setting, we first combine the 400 embeddings into a matrix $M$ by considering them as rows. Then, we apply SVD and low-rank approximation to obtain $\tilde{M}$. Finally, we compute the distance between $\tilde{M}$ and $M$ at each row. The smaller the distance, the more key information is kept by $\tilde{M}$. We experimented with different values for rank $r$. Figure 3 shows the Max and Mean distances between $M$ and in $\tilde{M}$. When the rank reaches 33 and above, the distance goes below 0.1 in average. Distance 0.1 suggests the two faces are very alike, as two images will be linked to the same person once their distance is below 1.1 under Facenet [52]. In other words, we can use a 33-dimensional embedding to approximate a 128-dimensional embedding without loosing much accuracy.

**Dimension reduction by EmbRev.** Though the adversary can use fewer queries to capture the key information of victim's face, how to solve the corresponding Equation 2 (when L2 distance is used) is unclear. Now the equation has infinite roots, as the number of equations ($m$) in Equation Set 1 is less than the number of the unknown elements ($n$) in $\vec{e_s}$. However, the adversary can choose to recover the "compressed" embedding directly by adjusting Equation 2. Below we describe the approach.

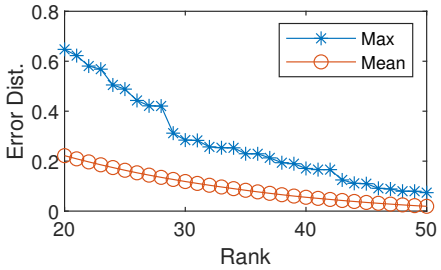


**Figure 3: Distances between $M$ and $\tilde{M}$ versus the rank of $\tilde{M}$ on Facenet-128.**

We assume $\vec{e_s}$ can be compressed into $m$ dimensions ($m < n$). With SVD, $\vec{e_s} = \vec{e_s^m} \cdot \Sigma_m \cdot V_m^\intercal + \delta$, in which $\Sigma_m$ is $m$-by-$m$, $V_m$ is $n$-by-$m$ and $\delta$ is the distance (or compression error) to $\vec{e_s}$. $\delta$ is usually quite small based our above analysis. By putting $\vec{e_s} = \vec{e_s^m} \cdot \Sigma_m \cdot V_m^\intercal + \delta$ into Equation 2, we obtain Equation 10.

$$\vec{e_s^m}^\intercal \cdot \Sigma_m \cdot V_m^\intercal \cdot V_m \cdot \Sigma_m \cdot \vec{e_s^m} + A \cdot V_m \cdot \Sigma_m \cdot \vec{e_s^m} + D + \Delta = 0 \quad (10)$$

where $A = -2 \cdot \{\vec{e_1}, \vec{e_2}, ..., \vec{e_m}\}^\intercal$ and $D = \{\vec{e_1}^\intercal \cdot \vec{e_1} - d_1^2, \vec{e_2}^\intercal \cdot \vec{e_2} - d_2^2, ..., \vec{e_m}^\intercal \cdot \vec{e_m} - d_m^2\}^\intercal$; $\Delta$ is the components with $\{\delta_1, \delta_2, ..., \delta_m\}$ where $|\Delta| \ll |D|$. All vectors are column vectors.

The issue of infinite roots does not exist in Equation 10 when we are solving $\vec{e_s^m}$, as $\vec{e_s^m}$ only has $m$ unknown elements and the rank of *any* matrix in Equation 10 cannot exceed $m$ ($V_m^\top \cdot V_m$ is the $m$-by-$m$ identity matrix and $A \cdot V_m$ is also $m$-by-$m$). Therefore, there are at most two roots, represented as Equation 11.

$$\vec{e_s^m} = (V_m \cdot \Sigma_m)^{-1} \cdot -A^{-1} \cdot (D + \Delta + \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}) \quad (11)$$

where $a$, $b$, $c$ are defined the same as Equation 8.

With $\vec{e_s^m}$ learnt, we compute $\vec{e_s}' = \vec{e_s^m} \cdot \Sigma_m \cdot V_m^\top$ ($\delta$ and $\Delta$ are neglected). Comparing to computing Equation 8, the only extra effort the attacker has to make is applying pseudo-inverse operation [62] on $A$ to get $A^{-1}$, as $A$ is not square. SVD is only implicitly used because $V_m$ and $\Sigma_m$ are eliminated when computing $\vec{e_s}'$.

For the later stage of face recovery, the slightly imprecise $\vec{e_s}'$ will be used as the input. Fortunately, if the compression error is negligible, we found the accuracy of the later stage is still high. As shown in Section 5.1, for the 128-dimensional Facenet model, with 60 queries, an attacker can recover the embedding of a victim with negligible error, which can further produce a clear face image that is similar to the version with 128 queries.

When Cosine distance is used, $\vec{e_s}'$ can be generated similarly under Equation 9. We skip the details.

### 3.3 EmbRev under No-box Setting

Under this setting, neither $f$ nor its embeddings are known to the adversary, so Equation 8 and 11 cannot be solved. Yet, such limitation can be addressed through attacking another embedding model $f'$ . Assume $f', f \in \mathcal{F}$, which is the function space of embedding models, and the accuracy of $f'$ and $f$ are similar. For images $\mathbf{x}$ and $\mathbf{y}$ drawn from the data distribution $p_{\text{data}}$, $f'$ and $f$ should derive the similar distance between any pair with high probability. In other words, $-\epsilon < \mathbb{E}_{\mathbf{x},\mathbf{y}\sim p_{\text{data}}}[||f(\mathbf{x}) - f(\mathbf{y})|| - ||f'(\mathbf{x}) - f'(\mathbf{y})||] < \epsilon$, where $\epsilon$ is a small positive number. When the adversary uses her own $f'$ to calculate $\vec{e_1}, \vec{e_2}, ..., \vec{e_m}$, the roots for $\vec{e_s}'$ or $\vec{e_s^m}$ will be similar with $f'(x)$ instead of $f(x)$. When attacking real-world FVS, the adversary can fine-tune $f'$ with the displayed similarity scores. To notice, $f$ and $f'$ do not need to have the same dimension number $n$ or even the same distance metric.

## 4 IMAGE RECOVERY

This module (called **ImgRev** by us) reconstructs victim's image from the inferred victim's embedding under the design of GAN, which has been overviewed in Section 2.3. Figure 4 shows the framework of **ImgRev** and it mainly consists of a novel embedding-to-image generator, a discriminator and loss functions.

**Overview. ImgRev** has a prominent difference comparing to existing GAN research in that we use **the embeddings instead of randomly generated noises as the generator's input**, and we call this method *embedding-reverse GAN (or erGAN)*. Before training, a set of realistic face images ($x \sim p_r(x)$ where $p_r$ is the data distribution over real samples) need to be collected to produce the embeddings ($e = f(x)$ where $f$ is the embedding model) for erGAN. As our evaluation shown in Section 5, using a *public* face dataset, like CelebA [38], is sufficient. The generator reconstructs images

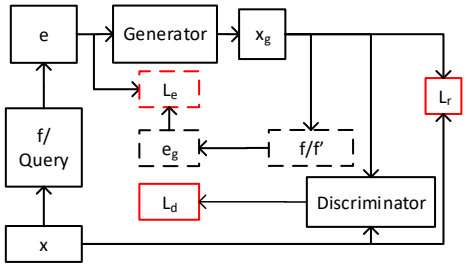


**Figure 4: Overview of ImgRev. Training images ($x$) are firstly converted by the embedding model ($f$) into embeddings $e$. The embeddings will be used by the generator to reconstruct images ($x_g$). The images will be used to compute losses ($L_r$, $L_d$ and $L_e$) and update the generator.**

($x_g = G(e)$, where $G$ is the generator) from the input embeddings $e$. Three kinds of loss will be used to direct the update of the $G$, which are 1) the recovery loss ($L_r$) that measures the recovery error at pixel level on the image plane; 2) the embedding loss that measures the recovery error on the embedding plane; 3) the discriminator loss that is inherited from the standard GAN, which measures if the distribution of $x_g$ falls into the distribution of $p_r$.

We follow the regular GAN training process [22], *i.e.*, training generator and discriminator in turns. The learning rate is decayed 0.02 for every epoch. The batch size is set to be 64. We train the generator 5 times after every single discriminator training iteration, which achieves good balance between the generator and the discriminator. After training, the generator of erGAN will be employed for image recovery and the discriminator will no longer be used. To notice, in this stage, *the adversary does not query the FVS under any setting (whitebox, blackbox and no-box setting).*

### 4.1 Generator

Ordinary GAN has generalization capability over noise field. It can generate realistic image but it has no control over image attributes. However, in our setting, we need the generated images to be tied to their input embeddings. Conditional GAN (cGAN) [20] has generalization capability over the noise field under the constraint of the label. If we regard embeddings as labels, cGAN indeed can make output images corresponding to embeddings. However, cGAN has no generality on the label, meaning that it can only generate images with seen labels, which does not satisfy our requirement. In contrast to the ordinary GAN and cGAN, erGAN has generalization capabilities *even over unseen embeddings*, *i.e.*, the embeddings recovered by **EmbRev**. Figure 5 illustrates the differences between different GAN methods at high level.

The generator of our erGAN has a multi-path phase and a single-path phase. Figure 6 shows the workflow of our generator for 512-dimensional embedding input. The first phase, *i.e.*, multi-path phase, extracts information from the input embedding at different paces. For the 512-dimensional embedding, the rapid branch directly deconvolutes the embedding from 512 dimensions to 512 10*10 tiny images. In contrast, the mild branch firstly deconvolutes it into tiny images of 2*2 then 10*10. These branches are combined together

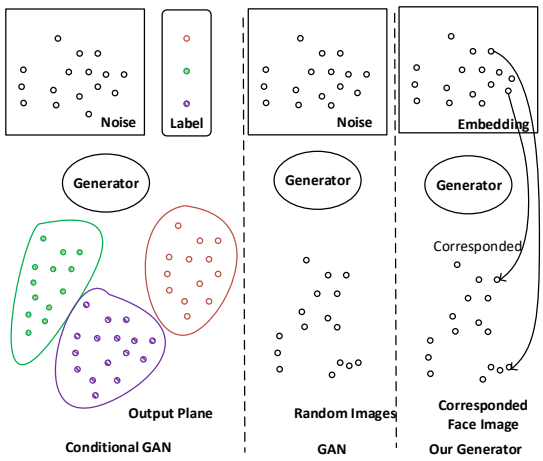


**Figure 5: Comparison of Conditional GAN generator, GAN generator and erGAN.**

after they reach the same size, providing a unified input for the later phase. The second phase, *i.e.*single-path phase, generates gradually clearer and larger images by concentrating channels. It repeatedly passes the deconvolution unit which enlarges the generated images by fusing multiple channels. During this stage, the size of the images is doubled while the channels are halved. The deconvolution unit is followed by a residual convolution unit [25] (see Figure 7) in order to rectify the images without changing the image size.

## 4.2 Discriminator

The discriminator tries to distinguish the generated images with the real face images to help the generator improve image quality. Our discriminator follows the design of the one used by WGAN-GP [23] (Wasserstein GAN with Gradient Penalty), which addresses the issue of training instability of GAN while producing high-quality images. WGAN-GP needs to maintain a Lipschitz function [48] to calculate the Wasserstein distance. It penalizes gradient for every independent sample. The discriminator we use drops all batch normalization layers (BN), and after every convolutional operation, we add a small Residual Block just like our generator, to avoid that the generator dominates the process. At last, the output of the discriminator will be a scalar value that is the confidence level that the discriminator considers $x_g$ falling within $p_r$.

## 4.3 Loss

Our loss function aggregates three types of losses and it can be represented as

$$L = w_r \cdot L_r + w_d \cdot L_d + w_e \cdot L_e \tag{12}$$

where $w_r$, $w_d$ and $w_e$ are weights. Through empirical analysis, we found 3:1:1 to 2:1:1 is the best ratio for $w_r$, $w_d$ and $w_e$, which encourages the recovered image to have realistic looking. Below we elaborate each loss.

**Discriminator loss ($L_d$).** We use the loss of WGAN-GP's discriminator for $L_d$ [23], represented as

$$L_d = \mathop{\mathbb{E}}_{x_g \sim p_g}[D(x_g)] - \mathop{\mathbb{E}}_{x \sim p_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x} \sim p_{\hat{x}}}\left[\left(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1\right)^2\right] \tag{13}$$

This loss tries to measure the difference between two distributions $p_g$ and $p_r$, *i.e.*, the generated images and the real images in our case. $D$ is the discriminator. $p_{\hat{x}}$ is the distribution of points uniformly sampled from the straight line between $p_g$ and $p_r$ and $\hat{x} = \epsilon x + (1-\epsilon)x_g$.

**Recovery loss ($L_r$).** To encourage the generator to produce realistic images, we add a loss item $L_r$ to force the generated image $x_g$ similar to the original image $x$. The loss penalizes the generator according to the pixel value difference between $x$ and $x_g$. Equation 14 shows $L_r$.

$$L_r = \mathop{\mathbb{E}}_{x \sim p_r, x_g \sim p_g}[||x - x_g||_1] \tag{14}$$

Here we use L1 distance (or Manhattan Distance) to measure the loss instead of L2 distance because we found L2 is more sensitive to the background part of images. In fact, two profile images usually differ more in background part than that in the face part. When calculating L2 distance, the larger difference, *i.e.*the background part takes dominant weight, as it is squared. To encourage the generator to focus on the face part, we choose L1 distance.

**Embedding loss ($L_e$).** We send the generated image ($x_g$) to a face embedding model ($f$) to get its embedding ($e_g$) and use the embedding loss ($L_e$) to penalize the difference between $e_g$ and the original embedding $e$, as shown by Equation 15.

$$L_e = \mathop{\mathbb{E}}_{x \sim p_r, x_g \sim p_g}[||f(x) - f(x_g)||] \tag{15}$$

To be noticed is that $f$ of FVS is unavailable to the black-box adversary. For white-box adversary, $f$ is identical to the one used by FVS. For no-box adversary, another embedding model $f'$ is used as an alternative of $f$ of FVS, which is explained in Section 3.3. However, for black-box adversary, only the result of $f$ is known by the adversary and we discuss this scenario in the next subsection.

## 4.4 ImgRev Under Black-box Setting

In this setting, though $L_e$ can be calculated, $\nabla L_e$ ($\nabla$ is derivative) is unknown as we have no access to $f$, which prevents erGAN to be guided by a face embedding model. Although erGAN can still be trained without $L_e$, *i.e.*, setting $L_e$ to zero, she would get poorer results because of the missing guidance from a face embedding model. To address this issue, she can use another open-source model $f'$ with similar accuracy as $f$ to obtain $\nabla L_e$, even when $f'$ and $f$ may have different model structure, distance metrics, etc. In fact, as open-source models have achieved quite high accuracy, their embeddings can tell the distinction between profile images well. They can be used to push the generator to generate more similar images. In other words, $||f'(x_1) - f'(x_2)||$ is expected to be positively correlated with $||f(x_1) - f(x_2)||$, where $x_1, x_2 \in \mathcal{X}$. Therefore, decreasing $||f'(x_1) - f'(x_2)||$ will result in the decrease of $||f(x_1) - f(x_2)||$. Therefore, $f$ can be replaced by $f'$ in Equation 15. To notice, this setting is different from using open-source models for white-box adversary or no-box setting as the the goal of $f'$ here is to output $\nabla L_e$.

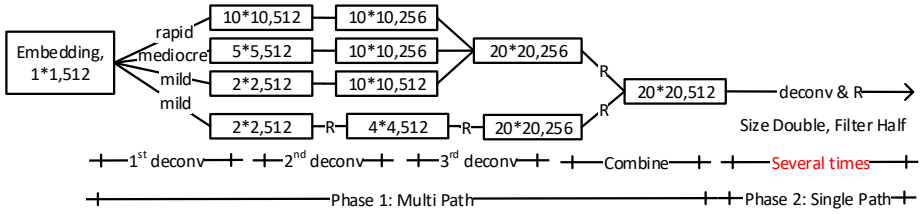


**Figure 6: The design of erGAN generator.**

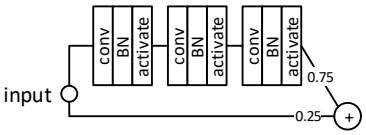


**Figure 7: Residual structure used by the generator.**

In rare cases when the adversary cannot find an open-source $f'$ with similar accuracy as $f$, she can train a surrogate model $f'$ through model extraction attack. In fact, previous works have demonstrated that model extraction by abusing public APIs of MLaaS models is feasible [13, 32, 32, 46, 58, 60]. Thousands of queries can produce a good surrogate model [32].

## 5 EVALUATION

In this section, we firstly evaluate **EmbRev**, focusing on the accuracy of the recovered embeddings. In addition, we also discussed the impact of query number, no-box setting, the quality of photos and the precision of displayed score. Then we evaluate **ImgRev**, focusing on how accurate the victims' faces can be recovered. As a highlight of our evaluation result, under whitebox setting, after the attacker issues 2 queries to an FVS using Facenet-128, **EmbRev** can reconstruct an embedding that bypasses FVS at **40%** chances. 20 queries guarantee **100%** success rate. With the recovered embedding, **ImgRev** is able to generate a discernible victim face (see Figure 10) without querying FVS. Below we elaborate the details.

**Targeted embedding models.** We examined the security of 4 embedding models with different embedding dimensions (128, 512, 1024, 1792) and distances (Cosine and L2). The widely used models like Facenet and Clarifai and an embedding model customized by us are tested. We adjust the distance threshold of each embedding model to match the accuracy reported by their literature or git repository using LFW dataset [35], because the threshold is not always public available. We are able to tune each model with the same or even better accuracy except Facenet. The reason is that we apply dlib [33] for alignment, following the design of OpenCV [12]. Table 1 shows the details of each embedding model.

For the customized embedding model, we built it on top of inception-resnet-v1 of Wide Residual Inception Network [1, 70]. Our customization includes adding cross-entropy loss over the "Additive Margin Softmax" layer after densing the embedding, which turns the model to a classifier for training. Because of this change, the embedding distance can be measured by Cosine distance. We trained the model with CASIA-Webface [68]. As shown in Table 1, moderate accuracy can be achieved.

**Experiment settings.** For the evaluation on **EmbRev**, we attack the two Facenet models. We tested the performance of **EmbRev** using LFW dataset and *no training is needed*. The white-box and black-box settings are jointly evaluated because they all allow the adversary to access the same score and embedding for each query. We evaluate the no-box setting separately by using another embedding model as surrogate model. For the evaluation on **ImgRev**, all 4 embedding models are attacked. We use celebA dataset to train and test **ImgRev**. We focus on white-box and black-box settings as the embeddings recovered under the no-box setting have large error margins. The black-box setting has result different from white-box as another open-source model $f'$ is leveraged to generate $\nabla L_e$.

For the overhead, the training of **ImgRev** to attack one embedding model costs us around 6 to 7 hours on a machine equipped with NVIDIA GeForce RTX 2080 Ti GPU, while recovering 32 face images as a batch in the testing stage costs 105 milliseconds. For **EmbRev**, the overhead is negligible.

| Model | Emb. Dim. | Distance Type | TH | Emb. Acc. |
|---|---|---|---|---|
| Residual Inception Network | 1792 | Cosine | 0.78 | 92.1% |
| Clarifai Online Face Embedding [10] | 1024 | Cosine | 0.55 | 98.1% |
| Facenet 20180402-114759 [51] | 512 | Cosine | 0.63 | 97.6% |
| Facenet 20170512-110547 [51] | 128 | L2 | 1.28 | 97.1% |

**Table 1: Embedding models evaluated by us. "Emb. Dim." is the embedding dimension. "TH" is the distance threshold below which two embeddings are considered to be of the same person. "Emb. Acc." is the accuracy of embedding model.**

### 5.1 Effectiveness of EmbRev

We used 300 photos from the LFW dataset to create the victim dataset. We sent each photo to the tested model and stored its embedding, which is the secret. Then, to simulate the attack, for each victim photo, we queried the tested embedding models with another set of photos (we call them query photos) and recorded all the embedding vectors and their distances to the victim photo. The distances and embeddings were inputted into **EmbRev** to recover the victim embedding. We implemented **EmbRev** with Matlab. When the number of queries equals to the embedding dimension (128 for Facenet-128), without exception, **every victim embedding can be recovered nearly perfectly**. The small error margins are caused by floating-point calculation, which are within $10^{-4}$ and far smaller than the threshold of embedding models.

**Reducing query number.** The analysis in Section 3.2 shows that 128-dimensional face embedding can be very close (*i.e.*, distance

less than 0.1) to its 33-dimensional compressed version, indicating that information inside face embedding is sparse. **EmbRev** makes full use of this property to reduce the number of queries issued by the adversary, so the attack can be even stealthier.

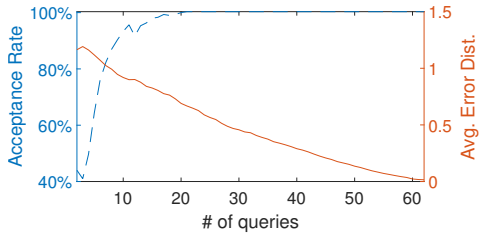


**Figure 8: Error Distances and acceptance rates versus the query numbers on Facenet-128.**

We firstly examined Facenet-128 model, by reducing the query number from 128 to 0 gradually. For each query number, we compute the error distance, *i.e.*, the distances between the recovered and victim embeddings, and show the average in Figure 8. It turns out even when we reduce the query number to half, *i.e.*, 60, the average error is very small (at 0.022). The error distance goes up to 0.1 when 53 queries are made which is still negligible as the distance threshold is 1.28 (shown in Table 1). The average error never exceeds 1.28, even with only two queries, in which case over 40% generated images can still be accepted by FVS. If only the attacker is able to make 20 queries, she has 100% chance to pass FVS. In Figure 3, we show the distances between $M$ (matrix of embeddings) and $\tilde{M}$ (lower-rank approximation of $M$) on Facenet-128. Given a query number $r$, the error distances introduced by $\tilde{M}$ at rank $r$ can be considered as its lower-bound. Through experiments, we found in order to reach such lower-bound, the attacker needs to query $r + 20$ times approximately.

Interestingly, when evaluating embedding models of higher dimensions, we found that the query number does not have to be increased. For Facenet-512, **EmbRev** costs the attacker only 39 queries to drop the mean error distance below 0.063 (10% of the threshold as shown in Table 1). We speculate it is because embedding models with better accuracy can extract more robust features, which can be captured by embedding models of lower dimensions.

**No-box setting.** For this experiment, we assume the targeted FVS uses Facenet-512, to which the adversary has no white-box or black-box access. She uses Facenet-128 model as the surrogate model to obtain embeddings and run **EmbRev**. To be noticed is that *Facenet-128 and Facenet-512 are very different embedding schemes*: L2 and Cosine distance are used respectively and they are trained on different datasets. The recovered embedding has 128 dimensions and we compute L2 distance under different query numbers. The result is presented in Figure 9.

Different from Figure 8, where error distance decreases following the increase of query number, Figure 9 shows error distance decreases first and then increases. The optimal result is observed when issuing 34 queries, where 1.026 is the average error distance. While the result is worse than the prior setting as expected, the adversary still has very good chance to bypass FVS.

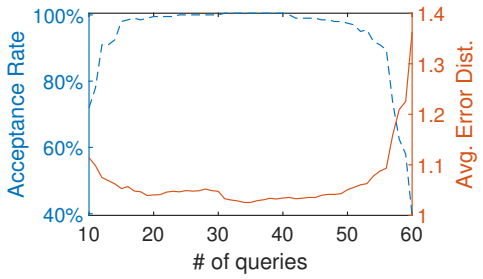


**Figure 9: Error distance and the acceptance rate versus the query numbers under no-box setting.**

For the following experiments with **ImgRev**, we neglect no-box setting as the error introduced from this step is still large enough (*i.e.*, over 1) that prevents victim face recovery. However, we consider **EmbRev** is effective under the no-box setting as the recovered embeddings can pass FVS (when issuing 34 queries).

**Precision of displayed score.** In the prior experiments, we assume the adversary can see the displayed score with high precision. However, the FVS operator or developer can choose to hide part of the score. For example, Figure 1a displays 16 digits while Figure 1b displays only 4 digits. When less digits are displayed, the embeddings recovered by **EmdRev** would be less accurate and we try to quantify this impact.

In particular, we truncate the distance values returned by the embedding models to 2 decimal fractions (*e.g.*, 1.23456 is truncated to 1.23) and re-run the experiment on Facenet-128 with 60 queries. The average error distance for this setting is 0.066 (in contrast, 0.022 for full precision), such error is well below the distance threshold. As FVS usually shows scores with at least 2 decimal fractions, **EmbRev** is shown to be robust against score truncation.

## 5.2 Effectiveness of ImgRev

We used the images from LFW dataset to test **EmbRev** but we found those images are not suitable for testing **ImgRev** as many of them were captured in unofficial occasions which would never been encountered by FVS. As such, we used another dataset, *celebA* [38], which consists of celebrity images labeled under 40 attributes, to train and test **ImgRev**. We remove the images with attributes of "Blurry", "Oval_Face" and "Bangs" and "Eyeglasses", because these images are taken usually not facing the camera with good angle or with coverings on faces. For FVS, photos usually have good angle and people do not wear coverings. The dataset was split into training and testing set of 20,480 ($DS_{train}$) and 1,800 ($DS_{test}$) images. To avoid the same person showing up in both datasets, we cluster the images based on their Identity ID and assign a cluster into either $DS_{train}$ or $DS_{test}$.

For each photo in $DS_{test}$ we generate its embedding using all 4 models listed in Table 1 and then use **ImgRev** to reconstruct the photo. Those embeddings can be considered as "perfect" embeddings recovered by the adversary. In the end of this section, we evaluate how errors produced by **EmbRev** impact the result of **ImgRev**.

We firstly tested the black-box settings without $L_e$ in loss function as the baseline. All four embedding models are tested. Then,

we tested white-box setting, where $\nabla L_e$ can be computed using the same embedding model $f$ as FVS and we use Facenet-128 as $f$. Finally, we tested the black-box setting again (Facenet-128 as $f$) but using another surrogate model $f'$ (Facenet-512) to generate $\nabla L_e$.

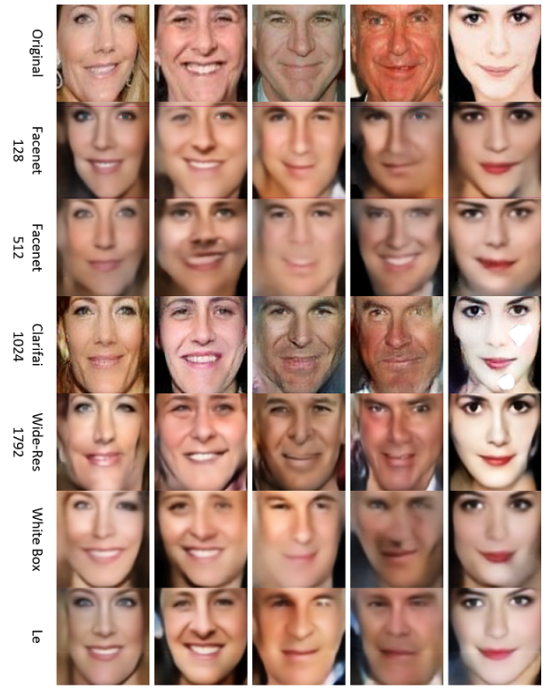


**Figure 10: First five samples in $DS_{test}$. The first row shows original photos. The second to the fifth row show the images recovered under blackbox baseline setting (without $L_e$). The sixth row shows white-box setting on Facenet-128. The last row shows the black-box setting with $L_e$ is generated under another model Facenet-512 ($f'$) when Facenet-128 is the targeted model ($f$).**

**Qualitative results.** We employ the trained generator to recover the first 1,800 images in $DS_{test}$ from their embeddings. Figure 10 shows the recovered versions of the first five images in $DS_{test}$. The victims can be easily discerned, suggesting **ImgRev** is quite effective. On the other hand, the recovery quality differs. **ImgRev** works best on Clarifai-1024, probably because Clarifai-1024 embeds more facial details, yielding more information to the adversary.

| | Blackbox Baseline | | | | White- | Blackbox |
|---|---|---|---|---|---|---|
| Model | 128 | 512 | 1024 | 1792 | box | $L_e$ |
| Acc. | 93.07% | 97.23% | 98.63% | 93.87% | 94.20% | 96.23% |
| FID | 114.11 | 157.47 | 33.94 | 49.39 | 86.00 | 61.25 |

**Table 2: The second row show the acceptance rate of the images recovered by ImgRev. The third shows the FID of the generated images (smaller is better). The settings are the same as Figure 10.**

**Quantitative results.** To quantify the attack effectiveness, we check the ratio of recovered images being accepted by FVS. Table 2 shows the acceptance rate, by comparing the original and recovered faces using the threshold defined in Table 1. As we can see, the quantitative results follow the general trend of qualitative results: Clarifai-1024 has the best acceptance rate, at 98.63%. Wide-Res-1792 has only 93.87% acceptance rate because the embedding is implemented by us, which has lower embedding accuracy. But even for the worst result on Facenet-128, **ImgRev** still achieves over 93% success rate.

In addition to acceptance ratio, we also compute FID (Frechet Inception Distance) [71] of each recovered image and report the average among them. FID records the distance between feature vectors calculated for real and generated images by GAN. Interestingly, though the acceptance rate is similar across embedding models, the difference is prominent under FID. Best performance is still achieved under Clarifai-1024. As FID of GAN generated images usually falls in the range from 30 to 200 [63], the image quality is acceptable.

One might argue that FVS operator can adjust the threshold to thwart our attack. To evaluate the effectiveness of this potential defense, we compute the embedding distances between images of 1) same person; 2) different persons and 3) original and recovered versions. Figure 11 shows the Probability Density Function (PDF) of the distances. It turns out for a victim photo, its distance to the photo recovered by **ImgRev** and other photos of the same person have similar distribution ("Recovered" curve and "Same" curve). Meanwhile, its distance to photos of other people ("Diff" curve) has very different distribution. Therefore, if this defense is applied to reject the photos provided by the adversary, false rejections will be significantly increased, making FVS unusable. Specifically, we evaluate the impact of FVS threshold on false-rejection and acceptance rate and show the result in Table 3. When the threshold is reduced to 0.4, where 35.84% of victim's verification requests are rejected, the attacker still has 48.96% success rate.

| TH | 0.7 | 0.6 | 0.5 | 0.4 | 0.3 |
|---|---|---|---|---|---|
| FR Rate | 4.49% | 8.79% | 18.75% | 35.84% | 61.72% |
| Acc. | 96.35% | 90.63% | 72.40% | 48.96% | 20.83% |

**Table 3: False-rejection rate and acceptance rate under different FVS thresholds.**

**Performance gain under whitebox Setting.** When the adversary knows the structure of the targeted embedding model, she can reliably compute $\nabla L_e$ and derive the embedding loss $L_e$, which should improve the quality of the recovered image. Here, we assess this expected performance gain. As listed in Table 2, the white-box setting brings to the attacker 1.2% gain of acceptance rate (94.20% compared to 93.07%) and 28.11 gain of image quality. Though such result shows white-box adversary has advantage over black-box adversary, the gain is small. *Therefore, for our attack to succeed, white-box access is not required.*

**Performance gain with surrogate model.** We evaluate if a black-box adversary can improve the baseline **ImgRev** by using an open-source surrogate model $f'$, which differs from $f$, to generate $L_e$.

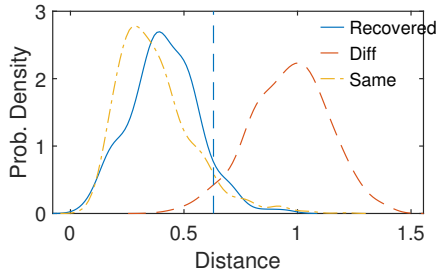


**Figure 11: PDF of distances between 1) embeddings of different photos under the same person ("Same"), 2) photos of different people ("Diff") and 3) recovered and original images ("Recovered"). The vertical line is the threshold for the Facenet-512 model (judge model).**


**Figure 12: Images Recovered with different number of queries.**

Surprisingly, the result shows that $f'$ brings to the attacker 3.2% acceptance rate gain and 52.86 gain of image quality, which are even better than the white-box setting. This somehow contradicts to our expectation, as white-box access should offer better insight into the targeted FVS.

After investigating the root cause, we found that such improvement might be caused by the higher verification accuracy of Facenet-512 compared to Facenet-128. $L_e$ generated by better model results in better recovery quality. In addition, the diversity brought by the surrogate model could help. With a different model supervising the image generation, it is more likely that the generator can generate images with fewer defects, because an embedding model may neglect certain features of an image which however are captured by another embedding model.

**Image recovery with imprecise embedding.** Our prior experiments assume the embedding recovered by the adversary is "perfect". Here we consider the embedding has error and evaluate **ImgRev** again.

In particular, we use **EmbRev** to recover the embeddings associated with $DS_{test}$ with different query numbers and then reverse those generated embeddings with **ImgRev**. The result confirms that **ImgRev** works well when the errors are small. When 60 queries are issued, the recovered images do not show obvious difference with the images recovered with 128 queries. Figure 12 shows the samples under different query numbers. The embeddings derived under photo distortion and score truncation lead to similar result (*i.e.*, 60 queries are sufficient) as the error margins introduced to embedding in these cases are even less (*e.g.*, $10^{-3}$ for query photo distortion).

## 6 DISCUSSION

While our research shows FVS can be bypassed and enrollees' privacy can be breached, limitations exist and are described below. In addition, we discuss the potential defense.

**Limitations.** 1) Under no-box setting, the embedding recovered by **EmbRev** is noisier comparing to other two settings. While the image recovered from the embedding is still able to bypass FVS, we found the image is dissimilar to victim's photo, hence we did not show it in the paper. But we want to point out that getting white-box or black-box access is feasible in most cases as FVS usually uses well-known embedding models. 2) We only evaluate the white-box attack scenario against Facenet-128, because the black box scenario already performs well and the improvement of $L_e$ is marginal. 3) The texture of images generated by **ImgRev** can be further improved. Images in Figure 10 show that coarse-gained features of victims' faces can be well recovered, like the outline, the position of eyes and nose, *etc.*. However, finer-grained features like skin textures are not well depicted, mainly because such information is not stored in an embedding. 3) We did not test our approach on the real-world FVS, like self-service FVS, due to ethical concerns. 4) We used a relatively small dataset to train and test the face embeddings and our attack. The result could differ when large dataset (*e.g.*, hundreds of millions of images are included). We acknowledge this limitation and plan to expand our evaluation with better hardware platform and more data. 5) We consider a "weaker authentication" scenario when liveness detection is not used.

**Defense.** Hiding the score (*e.g.*, only showing "pass/fail") is likely to solve this problem but it will make the on-site debugging much more difficult as described in Section 2.2. In fact, score is also encouraged to be shared on social media and many users are doing that [67]. Even when only "pass/fail" is shown, FVS is not bullet-proof as the adversary can issue more queries till discovering an embedding similar enough to victim's. Another approach is to add noises to the values visible to the attacker (*e.g.*, confidence score vector [31]), but false positives would rise against legitimate users.

ML library and SDK documentation should clearly tell developers that distances can only be exposed to authorized managers and can never be displayed to normal users. Developers should also learn case studies about embedding leakages so they will not leak distances inadvertently.

To thwart the image recovery attack after embedding is inferred, the embedding model can be redesigned to add privacy protection. Just like a one-way hash function, ML developers may design models in a way that the reverse mapping of a model cannot be easily figured out by attackers. Hash functions employ computation

subroutines that are hard to reverse. However, because basic units used by DNN today like pooling, convolution, activation, are all partially or totally reversible, making the embedding model irreversible would be unlikely to succeed. Therefore, new DNN units should be developed to fulfill this goal. In the meantime, auditing the queries and blocking the follow-up ones when the distribution is abnormal can be used to deter the model inversion attack [32] and we will investigate whether it can provide strong protection on embedding models.

## 7 RELATED WORKS

We review those relevant works in machine-learning field first. Then, we overview other works about face authentication.

**Data confidentiality.** Fredrikson *et al.* proposed model inversion attack (MIA) [19] and showed that the model used for medical treatment leaks patient's genetic markers. Following this work, Fredrikson *et al.* showed confidence values exposed by the prediction API of MLaaS can be exploited to reconstruct part of training data[18]. Specifically for their face recognition experiment, they recovered images of victims in the training set. Recently, Yang *et al.* improved the accuracy of image reconstruction of MIA using public auxiliary dataset [67].

To be noticed is that our work assumes a different scenario for image reconstruction, *i.e.*, face authentication. In the previous works, a vector of logits (e.g., confidence value or prediction scores) can be obtained by the adversary for each prediction. However, only one score is returned to our adversary, which does not reveal much information about the model's characteristics like gradient or special-featured gradient resulted from over-fitting. Yet, by exploiting the unique properties of face embedding, we found victims faces can be recovered.

A related task as ours is *image generation*, where encoder-decoder network [6, 41] has been used. As far as we know, the work done by Zhmoginov *et al.* [73] is the only one reconstructing image from embedding. However, as their goal is to transfer an image to another one such that it has close distance to an embedding (small distance in embedding plane), the generated image is dissimilar to victim's image (large distance in image plane).

In addition to MIA, previous works showed certain properties of the training data can be revealed. Reza *et al.* proposed membership inference attack [54]. Later, the same attack is demonstrated successful in other settings [24, 39, 42, 49].

**Model confidentiality.** By exploiting the prediction API of machine-learning models, researchers found the model structure (*e.g.*, hyper-parameters and weights) [9, 13, 32, 46, 58, 60, 69] and optimization procedure can be revealed [45]. In addition to exploiting the algorithm weakness of machine-learning models, researchers found the hardware executing them also leaks model structure through side channels. In particular, the performance counters provided by GPU [43], shared CPU cache [26, 65], electromagnetic signals [7], memory access patterns [27, 28], power [61] and execution time [15] can be exploited to this end. Previous works studied model confidentiality and data confidentiality in separate directions, but they might be able to augment each other (*e.g.*, knowing model structure could increase the accuracy of the data inference attacks). We will investigate how our attack can be facilitated with the help of inference attacks on model structure.

**Security of face authentication.** The major concern is that face verification can be fooled by replaying an image forged from victim's public photos. As such, most recent works involved liveness detection as the countermeasure [36, 57, 59] but researchers also discovered new attacks against it [64].

Recently, researchers showed that through generating adversarial physical example (e.g., eyeglass frames), face authentication can be fooled [53, 74]. While our attack can be categorized as adversarial learning, the adversary model is very different. Their attack assume victim's facial image has been possessed by the adversary so the adversarial example can be built upon it through perturbation, but our attack assumes zero knowledge about the victim's appearance.

A recent work proposed to use distance to assist GAN to generate adversarial examples [66], but they did not recover the enrollee's embeddings and images like ours.

## 8 CONCLUSION

Our study reveals that the small information leakage from face verification system (FVS), *i.e.*, the score displayed after each verification request, can be accumulated to recover victim enrollee's real face. By acquiring only a dozen of scores, she can readily recover the embedding of the victim's face, with our proposed embedding-recovery equations. What's worse is that the embedding is equally sensitive as the victim's face. As a proof, we designed a recovery model based on GAN to convert the recovered embeddings back to face images, the results show both embedding and face recovery are effective, as the FVS can be bypassed at high probability and the recovered face is similar to the victim.

## ACKNOWLEDGMENTS

The Fudan authors are supported by NSFC 61802068. The UCI author is partially supported by NSF DGE-2039634, and gift from Microsoft and Cisco.

## REFERENCES

[1] 2020. Wide Resnet Git. https://github.com/szagoruyko/wide-residual-networks. https://github.com/szagoruyko/wide-residual-networks Accessed: 2020-01-10.
[2] AMGTime. [n.d.]. Face Recognition, Fingerprint, Proximity Cards 4 in 1 Biometric Time Attendance Package. https://amgtime.com/hardware-facial-recognition-technology-rfid-time-attendance. Accessed: 2019-12-20.
[3] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. 2016. *OpenFace: A general-purpose face recognition library with mobile applications.* Technical Report. CMU-CS-16-118, CMU School of Computer Science.
[4] Apple. [n.d.]. About Face ID advanced technology. https://support.apple.com/en-us/HT208108. Accessed: 2019-12-20.
[5] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875* (2017).
[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* 39, 12 (2017), 2481–2495.
[7] Lejla Batina, Shivam Bhasin, Dirmanto Jap, and Stjepan Picek. 2019. CSI NN: Reverse Engineering of Neural Network Architectures Through Electromagnetic Side Channel. In *28th USENIX Security Symposium (USENIX Security 19).* USENIX Association, Santa Clara, CA, 515–532. https://www.usenix.org/conference/usenixsecurity19/presentation/batina
[8] Matthew Braga. [n.d.]. Facial recognition technology is coming to Canadian airports this spring. https://www.cbc.ca/news/technology/cbsa-canada-airports-facial-recognition-kiosk-biometrics-1.4007344. Accessed: 2019-12-20.
[9] Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. 2020. Exploring Connections Between Active Learning and Model

Extraction. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Boston, MA.

[10] Clarifai. 2019. Clarifai Face Embedding. https://www.clarifai.com/models/face-embedding-image-recognition-model-d02b4508df58432fbb84e800597b8959.

[11] Clarifai. 2020. Computer Vision AI Technology Case Studies. https://www.clarifai.com/customers. Accessed: 2020-04-10.

[12] Intel Corporation, Willow Garage, and Itseez. [n.d.]. OpenCV. https://opencv.org/. Accessed: 2019-01-20.

[13] Jacson Rodrigues Correia-Silva, Rodrigo F Berriel, Claudine Badue, Alberto F de Souza, and Thiago Oliveira-Santos. 2018. Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[14] DERMALOG. 2020. DERMALOG Home Page. https://www.dermalog.com/products/software/face-recognition/. Accessed: 2020-01-10.

[15] Vasisht Duddu, Debasis Samanta, D. Vijay Rao, and Valentina E. Balas. 2018. Stealing Neural Networks via Timing Side Channels. *CoRR* abs/1812.11720 (2018). arXiv:1812.11720 http://arxiv.org/abs/1812.11720

[16] Nesli Erdogmus and Sebastien Marcel. 2014. Spoofing face recognition with 3D masks. *IEEE transactions on information forensics and security* 9, 7 (2014), 1084–1097.

[17] Australian Border Force. [n.d.]. Smartgates. https://www.abf.gov.au/entering-and-leaving-australia/smartgates/arrivals. Accessed: 2019-12-20.

[18] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1322–1333.

[19] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing.. In *USENIX Security Symposium*. 17–32.

[20] Jon Gauthier. 2014. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester* 2014, 5 (2014), 2.

[21] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *Vldb*, Vol. 99. 518–529.

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

[23] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems*. 5767–5777.

[24] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2017. LOGAN: evaluating privacy leakage of generative models using generative adversarial networks. *arXiv preprint arXiv:1705.07663* (2017).

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[26] Sanghyun Hong, Michael Davinroy, Yiğitcan Kaya, Stuart Nevans Locke, Ian Rackow, Kevin Kulda, Dana Dachman-Soled, and Tudor Dumitraş. 2018. Security analysis of deep neural networks operating in the presence of cache side-channel attacks. *arXiv preprint arXiv:1810.03487* (2018).

[27] Xing Hu, Ling Liang, Lei Deng, Shuangchen Li, Xinfeng Xie, Yu Ji, Yufei Ding, Chang Liu, Timothy Sherwood, and Yuan Xie. 2019. Neural Network Model Extraction Attacks in Edge Devices by Hearing Architectural Hints. *CoRR* abs/1903.03916 (2019). arXiv:1903.03916 http://arxiv.org/abs/1903.03916

[28] Weizhe Hua, Zhiru Zhang, and G. Edward Suh. 2018. Reverse Engineering Convolutional Neural Networks Through Side-channel Information Leaks. In *Proceedings of the 55th Annual Design Automation Conference* (San Francisco, California) *(DAC '18)*. ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3195970.3196105

[29] Idency. [n.d.]. Facial Recognition Time and Attendance Machines. https://idency.com/product-category/authentication/time-and-attendance/biometric-time-and-attendance-systems/facial-time-attendance/. Accessed: 2019-12-20.

[30] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 5967–5976.

[31] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 259–274.

[32] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N Asokan. 2019. PRADA: protecting against DNN model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 512–527.

[33] Davis E. King. [n.d.]. dlib C++ library. http://dlib.net/. Accessed: 2019-01-20.

[34] Andrea Lagorio, Massimo Tistarelli, Marinella Cadoni, Clinton Fookes, and Sridha Sridharan. 2013. Liveness detection based on 3D face shape analysis. In *2013 International Workshop on Biometrics and Forensics (IWBF)*. IEEE, 1–4.

[35] Gary B. Huang Erik Learned-Miller. 2014. *Labeled Faces in the Wild: Updates and New Reporting Procedures*. Technical Report UM-CS-2014-003. University of Massachusetts, Amherst.

[36] Yan Li, Yingjiu Li, Qiang Yan, Hancong Kong, and Robert H. Deng. 2015. Seeing Your Face Is Not Enough: An Inertial Sensor-Based Liveness Detection for Face Authentication. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. ACM, New York, NY, USA, 1558–1569. https://doi.org/10.1145/2810103.2813612

[37] Yan Li, Ke Xu, Qiang Yan, Yingjiu Li, and Robert H Deng. 2014. Understanding OSN-based facial disclosure against face authentication systems. In *Proceedings of the 9th ACM symposium on Information, computer and communications security*. ACM, 413–424.

[38] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[39] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. 2018. Understanding Membership Inferences on Well-Generalized Learning Models. *CoRR* abs/1802.04889 (2018).

[40] Jaime Lorenzo-Trueba, Fuming Fang, Xin Wang, Isao Echizen, Junichi Yamagishi, and Tomi Kinnunen. 2018. Can we steal your vocal identity from the Internet?: Initial investigation of cloning Obama's voice using GAN, WaveNet and low-quality found data. *arXiv preprint arXiv:1803.00860* (2018).

[41] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*. 2802–2810.

[42] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting Unintended Feature Leakage in Collaborative Learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*. 691–706. https://doi.org/10.1109/SP.2019.00029

[43] Hoda Naghibijouybari, Ajaya Neupane, Zhiyun Qian, and Nael Abu-Ghazaleh. 2018. Rendered insecure: GPU side channel attacks are practical. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2139–2153.

[44] NPR. [n.d.]. Police Facial Recognition Databases Log About Half Of Americans. https://www.npr.org/2016/10/23/499042369/police-facial-recognition-databases-log-about-half-of-americans. Accessed: 2019-12-20.

[45] Seong Joon Oh, Max Augustin, Bernt Schiele, and Mario Fritz. 2017. Towards reverse-engineering black-box neural networks. *arXiv preprint arXiv:1711.01768* (2017).

[46] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. ACM, 506–519.

[47] Jeff John Roberts. [n.d.]. Here's How Many Adult Faces Are Scanned From Facial Recognition Databases by Cops. https://fortune.com/2016/10/18/facial-recognition-database/. Accessed: 2019-12-20.

[48] Todd Rowland and Eric W Weisstein. [n.d.]. Lipschitz Function. http://mathworld.wolfram.com/LipschitzFunction.html. Accessed: 2020-01-02.

[49] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).

[50] Samsung. [n.d.]. How does Face recognition work on Galaxy Note10, Galaxy Note10+, and Galaxy Fold? https://www.samsung.com/global/galaxy/what-is/face-recognition/. Accessed: 2019-12-20.

[51] David Sandberg. 2019. The most popular facenet implementation and pre-trained model. https://github.com/davidsandberg/facenet. Accessed: 2019-01-20.

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

[53] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. ACM, New York, NY, USA, 1528–1540. https://doi.org/10.1145/2976749.2978392

[54] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 3–18.

[55] Avinash Kumar Singh, Piyush Joshi, and Gora Chand Nandi. 2014. Face recognition with liveness detection using eye and mouth movement. In *2014 international conference on signal propagation and computer technology (ICSPCT 2014)*. IEEE, 592–597.

[56] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html


**Figure 13: (a) and (b) are generic male and female images. (c) is the image with the targeted embedding. (d) and (e) are transformed from (a) and (b).**


[57] Di Tang, Zhe Zhou, Yinqian Zhang, and Kehuan Zhang. 2018. Face Flashing: a Secure Liveness Detection Protocol based on Light Reflections. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_03B-5_Tang_paper.pdf

[58] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs.. In *USENIX Security Symposium*. 601–618.

[59] Erkam Uzun, Simon Pak Ho Chung, Irfan Essa, and Wenke Lee. 2018. rtCaptcha: A Real-Time CAPTCHA Based Liveness Detection System.. In *NDSS*.

[60] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing Hyperparameters in Machine Learning. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. 36–52.

[61] Lingxiao Wei, Bo Luo, Yu Li, Yannan Liu, and Qiang Xu. 2018. I Know What You See: Power Side-Channel Attack on Convolutional Neural Network Accelerators. In *Proceedings of the 34th Annual Computer Security Applications Conference* (San Juan, PR, USA) *(ACSAC '18)*. ACM, New York, NY, USA, 393–406.

[62] Wolfram. [n.d.]. Pseudoinverse. http://mathworld.wolfram.com/Pseudoinverse.html. Accessed: 2019-01-20.

[63] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. 2018. Elegant: Exchanging latent encodings with gan for transferring multiple face attributes. In *Proceedings of the European conference on computer vision (ECCV)*. 168–184.

[64] Yi Xu, True Price, Jan-Michael Frahm, and Fabian Monrose. 2016. Virtual u: Defeating face liveness detection by building virtual models from your public photos. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 497–512.

[65] Mengjia Yan, Christopher Fletcher, and Josep Torrellas. 2018. Cache telepathy: Leveraging shared resource attacks to learn DNN architectures. *arXiv preprint arXiv:1808.04761* (2018).

[66] Lu Yang, Qing Song, and Yingqi Wu. 2021. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network. *Multimedia Tools and Applications* 80, 1 (2021), 855–875.

[67] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. 2019. Neural Network Inversion in Adversarial Setting via Background Knowledge Alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. ACM, New York, NY, USA, 225–240. https://doi.org/10.1145/3319535.3354261

[68] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. 2014. Learning Face Representation from Scratch. *CoRR* abs/1411.7923 (2014). arXiv:1411.7923 http://arxiv.org/abs/1411.7923

[69] Honggang Yu, Kaichen Yang, Teng Zhang, Yun-Yun Tsai, Tsung-Yi Ho, and Yier Jin. 2020. CloudLeak: Large-Scale Deep Learning Models Stealing Through Adversarial Examples. In *NDSS*.

[70] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016).

[71] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. 2018. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318* (2018).

[72] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 4006–4015.

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

[74] Zhe Zhou, Di Tang, Xiaofeng Wang, Weili Han, Xiangyu Liu, and Kehuan Zhang. 2018. Invisible mask: Practical attacks on face recognition with infrared. *arXiv preprint arXiv:1803.04683* (2018).

[75] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2242–2251.


# APPENDIX

## A PHOTOS GENERATED BY [73]

The photos shown in Figure 13 are taken from the paper of Zhmoginov *et al.* [73]. (d) and (e) are the reconstructed images whose embeddings are close to the embedding of (c), but they are dissimilar to (c) on the image plane. In contrast, **ImgRev** is able to produce image similar to the targeted person.