Inverting Biometric Models with Fewer Samples: Incorporating the Output of Multiple Models

Sohaib Ahmad
University of Connecticut
Storrs
sohaib.ahmad@uconn.edu

Kaleel Mahmood
University of Connecticut
Storrs
kaleel.mahmood@uconn.edu

Benjamin Fuller
University of Connecticut
Storrs
benjamin.fuller@uconn.edu

2022 IEEE International Joint Conference on Biometrics (IJCB) | 978-1-6654-6394-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCB54206.2022.10007997

Abstract

Authentication systems are vulnerable to model inversion attacks where an adversary is able to approximate the inverse of a target machine learning model. Biometric models are a prime candidate for this type of attack. This is because inverting a biometric model allows the attacker to produce a realistic biometric input to spoof biometric authentication systems.

One of the main constraints in conducting a successful model inversion attack is the amount of training data required. In this work, we focus on iris and facial biometric systems and propose a new technique that drastically reduces the amount of training data necessary. By leveraging the output of multiple models, we are able to conduct model inversion attacks with 1/10th the training set size of Ahmad and Fuller (IJCB 2020) for iris data and 1/1000th the training set size of Mai et al. (Pattern Analysis and Machine Intelligence 2019) for facial data. We denote our new attack technique as structured random with alignment loss.

1. Introduction

Many authentication systems are based on biometric identification [1, 2]. Two widely adopted biometrics include iris and facial recognition. Despite the prevalence of these biometric based authentication systems, they remain vulnerable to a type of attack called a model inversion [3]. In a model inversion attack, an adversary is able to train an attack model that approximates the inverse of the target biometric model used in the authentication system. Once the adversary is able to succeed in training this attack model, they are able to produce realistic looking biometrics. These realistic looking biometrics can be used for spoofing attacks [4], where an attacker creates a "fake" version of a user's biometric.

Deep learning models are increasingly being used for biometrics [5–10]. Fredikson et al. initiated model inversion attacks on such networks, targeting the facial biometric [3]. Recent model inversion attacks use generative adversarial networks or GANs [11] and use auxiliary information such as blurred faces.

To set notation, denote the trained biometric identification system as f_T to indicate it is the model being targeted in the attack. The attack proceeds in stages:

Training The attacker receives ℓ samples of the form

$$(x_i, y_i = f_T(x_i)).$$

At the end of this stage the attacker outputs a model f_T^{-1}. It should be the case that for unseen pairs x', y' the value $f_T^{-1}(y')$ is similar to x'.

Test/Attack The attacker receives values y' and inverts them to produce realistic biometric values $f_T^{-1}(y')$.

A limitation of prior work is the need for a large number of training samples. Mai et al. [12] require 2×10^6 training samples in their attack on the facial biometric. Ahmad and Fuller [13] require 2×10^4 training samples in their attack on the iris biometric. While large facial and iris datasets exist, model inversion targets smaller applications. It is thus crucial to determine if model inversion is possible with fewer training points.

We investigate whether the adversary can substitute the output of multiple models in Training in place of more training samples.[1] We consider the following new attack setup (for parameter α):

Training Let $f_{T_1}, \ldots, f_{T_\alpha}$ be models used in training a final model f_{T_α}. The attacker receives ℓ samples of the form

$$(x_i, f_{T_1}(x_i), f_{T_2}(x), \ldots., f_{T_\alpha}(x_i)).$$

At the end of this stage the attacker outputs a model $f_{T_\alpha}^{-1}$. It should be the case that for unseen pairs x', y' the value $f_{T_\alpha}^{-1}(y')$ is similar to x'.

[1] Salem et al. [14] study the difference a model undergoes when it is updated in an online fashion. Their work considers small updates while we explore larger changes when the target model's dataset undergoes deletion or addition of classes.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

Test/Attack The attacker receives values y' and inverts them to produce realistic biometric values $f_{T_\alpha}^{-1}(y')$.

Multiple works have considered attack avenues to steal models [15–17]. We review three settings when multiple models are available in Section 2.1. We ask whether an attacker who sees the output of multiple models when training the attack model is able to invert more effectively. The research question of this work is:

> How to effectively use multiple models to reduce the training set size?

We consider training set size of $\ell = 2 \times 10^3$. Mai et al. [12] used $\ell = 2 \times 10^6$, Ahmad and Fuller [13] used $\ell = 2 \times 10^4$.

Our attacks are performed on raw templates which are output from biometric networks and stored insecurely. There are two relevant lines of work on securing biometric models. One line shows how to encrypt the output of biometric networks [1, 18–31] in a way that authentication systems still work. These methods have constraints where the provided security (in bits) is small or authentication is slow. A second line show how to securely train models and allow these models to be evaluated privately [32–34]. Our attacks are black box but do need the ability to observe f_{T_i} for multiple i.

1.1. Attack Approach

The high level architecture of our inversion attack is a generative adversarial network or GAN [35] as in prior work on biometric model inversion [12, 13]. A GAN is a pair of algorithms, a generator and a discriminator. In usual image applications, the generator takes random noise. The generator's goal is to produce images that the discriminator cannot distinguish from true training samples. As with previous work [12, 13], we modify this paradigm, making the GAN generator take the output of biometric transform as input. The discriminator is then given either real biometrics or those created by the generator. By fooling the discriminator, the generator works as our attack model and an inverter for the biometric transform. Yang et al. [36] proposed a simple mechanism for incorporating multiple models:

Random During attack model training, a random $1 \le i \le \alpha$ is selected and the pair $(x_j, f_{T_i}(x_j))$ is provided as ground truth for the GAN.

We show visual reconstructions of irises in Figure 3, deferring discussion of results and visual reconstructions of faces until Section 5. The Random or Rand method does recover the high level shape of the iris but is missing crucial details such as 1) a crisp boundary between the iris and the pupil and 2) iris texture.

1.2. Our Contribution

Let m denote the output dimension of f_{T_i}. Yang et al. [36] consider a GAN with input length of m. All of our new methods consider a GAN takes inputs of length $\alpha \cdot m$. We call these networks input-augmented GANs. We introduce the following input-augmented GANs:

Concatenation In this approach the GANs training samples are the entire tuple $(x_j, f_{T_1}(x_j), f_{T_2}(x), \ldots, f_{T_\alpha}(x_j))$.

Structured Random Sample a random $1 \le i \le \alpha$ as above and set other components of the vector to 0. That is, input

$$(x_j, 0, 0, \ldots, 0, f_{T_i}(x_j), 0, \ldots, 0).$$

Structured Random w/ Alignment Loss This approach follows the structured random approach above, but also asks the GAN to predict i.

Going forward we refer to these three methods of incorporation as Concat, SR and SRwAL respectively. We consider two types of target models: feature extractors and classifiers (see Section 4.4).

SRwAL provides the best results see Section 6). This is interesting in comparison to SR because the only difference is that SRwAL asks the model to remember which location i is nonzero. Even though the value i is "easy" to predict, forcing the GAN to predict this value improves overall performance. We believe that the GAN is better able to distinguish between inputs from different models, which leads to better inversion on the final model f_{T_α}. Our accuracy results are in Tables 1 and 2.

Organization The rest of this work is organized as follows: Section 2 describes the system architecture, Section 3 reviews how feature extractors and classifiers are used in biometrics, Section 4 describes our attack model, Sections 5, 6 present evaluation methodology and results respectively. Section 7 concludes.

2. Adversarial Model

This section describes the adversarial model. We defer discussion of measuring attacker success until Section 4.4. Recall that we use x to denote the input to the target network and $y = f_T(x)$ to indicate the resulting output. The goal of the attack is to train a network f_T^{-1} that on input y that can predict x. As mentioned in the Introduction, we assume that the adversary has access to the output of multiple related models. That is, in the Training stage they receive tuples of the form

$$x_i, f_{T_1}(x_i), f_{T_2}(x), \ldots, f_{T_\alpha}(x_i).$$

the goal of the training stage is to produce a model $f_{T_\alpha}^{-1}$ where it is true that

$$f_{T_\alpha}^{-1}(f_{T_\alpha}(x')) \approx x'.$$

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

We use i to additionally index the target model, for example: $x, \{f_{T_i}(x) = y_i\}_{i=1}^{\alpha}$. The parameter α controls how many models the adversary has access to. The second stage of the attack is denoted as Test where we assume outputs $f_{T_\alpha}(x')$ leak and the attacker will reconstruct x'.

2.1. Accessing Multiple Models

We consider three types of related models that may be available to an attacker that we call Upslope, Update and Downslope.

Upslope In the first setting, we consider the intermediate models that are created when a model is first trained. Due to the complexity of modern models, training is a computationally intensive process and is done in epochs. Since training is a complex, error-prone process models and performance data are stored for debugging purposes.[2] The target iris and face recognition models converge in 100 epochs (see Section 5.1).

We utilize five different models saved during training in this attack. We utilize models after 0 (Pre-trained on ImageNet), 25%, 50%, 75% and 100% of training. At attack time, we consider two settings when only the final production model's output is available and when all models are available. The setting when all models are available at test time is used to compare the different methods for incorporating multiple vectors and is not intended to be realistic.

Update Addition of a new user into the system that needs to be learned by the model. In this case the attacker may be able to prepare the images used in training the model on the new user. That is, the image need not come from the honest biometric distribution. A natural setting in which an attacker could perform Upslope and Update attacks is federated learning [33, 37]. In this setting, a model is trained but the adversary asks the model to learn on new images.

Images in the Update attack are crafted by the adversary. The updates are crafted by taking a normal biometric image and applying a Gaussian blur using a 3x3 kernel and sigma of 0.8 to all images of the new user being added. Blurring makes images from different classes appear similar. Fredrikson et al. [38] perform a similar attack where they recover original faces from blurred out images however we perform a smaller amount of blur. We retrain the target model for 10 epochs. The attacker has access to the original, 5th and 10th model.

Downslope Removal of a person from the system. Such a removal may occur due to right to be forgotten legislation which has resulting in the field of machine unlearning [39–41]. In this setting, the adversary requests an individual be removed but has no control over how this removal is processed. Recent laws and regulations have also taken privacy risks into account. The General Data Protection Regulation (GDPR) [42] in the European Union and the California Consumer Privacy Act (CCPA) [43] in the United States call for more action to protect personal data and control how and where data is stored. In addition to simplifying rules on data storage and privacy, this legislation grants control to a person over their personal data, consequently, a person can ask a company to remove their data.

We assume machine unlearning is performed naively: completely retraining the model after deleting required data from the training dataset. Attacking more sophisticated unlearning strategies is an important piece of future work.

We retrain the target model (to perform unlearning) for 100 epochs on the new training dataset. We utilize models after 0, 25%, 50%, 75% and 100% of re-training has been completed for a total of 5 models. We assume the adversary removes multiple people/classes from the training set of a model. We remove 10 classes from our iris application and 5 classes from our face dataset and then retrain the model.

For all attacks except for the Update attack, the adversary can passively receive normal biometric images and their corresponding outputs. As mentioned above for the Update attack, these images are prepared specially and differ from the normal biometric distribution.

In our attacks we only assume the output of the model, either a template or a classification vector, is revealed. This is in contrast to models that assume knowledge of the internal weights of the models f_{T_i} such as Fredriskson et al. [38].

3. Review of Types of Biometric Target Models

Feature extractors $f_{E,T}$ Feature extraction networks [44] output a m-dimensional feature vector. Feature extractors are trained to generate embeddings from given biometric inputs. As an example, given a biometric input x a network $f_{E,T}$ generates an embedding/feature vector $y = f_{E,T}(x)$. This embedding is known as a template and stored in a database (in mobile devices this storage is inside of a secure enclave). At a subsequent reading of the same biometric input (with noise) another embed-ding $y' = f_{E,T}(x')$ is produced. A distance metric such as Euclidean distance is used to compare the two vectors $d = L_2(y, y') = \sqrt{\sum_{i=1}^{n}(y_i - y_i')^2}$. The biometric will authenticate an individual successfully if the distance is below a precomputed threshold, denoted $thres$. Training feature extraction networks generally does not require a set number of classes, only labelling which samples should be grouped together or pushed apart. Feature extractors are used in applications when not all users are known when the model is trained.

Classifiers $f_{C,T}$ Classification networks output an m-dimensional classification vector. Classifiers work with known number of classes. The objective is to learn a clas-

[2] Salem et al. [14] considered the related question is whether the difference in two models when an individual item is added leaks about that individual item.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

sification vector such that every input x that belongs to a class from {1, . . . , m} is assigned to its class in the classification vector. Usually, the output layer of a classification network is a softmax based layer which takes the preceding (feature vector) layer and maps it to a classification vector. When all users are known at training time, classifier use for biometric identification is straightforward. A biometric is deemed to belong to class i if the classification output indicates membership in class i with high enough confidence (which depends on the application).

The second last layer of a classification network is a fully connected layer and serves as a high quality feature extractor.

4. Attack Model Design

4.1. Attack Goal

As a reminder, for a target network f_T where $y' = f_T(x')$ the goal is to learn a transform f_T^{-1} such that for $x^{\square} = f_T^{-1}(y')$ the values $f_T(x')$ and $f_T(x^{\square})$ are similar.

We briefly review the goals for the two settings of feature extractors and classifiers. For feature networks the goal is given y to produce an x' such that $x' \approx x$ where $y = f_T(x)$. In classification networks, the goal is to produce an x' that will be classified as class i with the highest confidence possible [45]. As we are using a GAN to produce these x' there is a secondary goal that x' appears similar to valid x. This may not be the case if x' was simply the class average [36].

The reason for the difference in goal is because of the difference in how these network types are used in identification systems. Feature extractors and classifiers are used differently in identification systems. Feature extractors are used to extract templates that are compared with a stored value. Thus, the goal is to be able to recreate the stored template as accurately as possible. Classifiers judge an input to be in a class if has "high enough" confidence of being assigned to that class so the goal is simply to maximize that confidence. Thus, the attacker goal in both settings to produce an image that will authenticate with the highest probability. In the literature the feature extractor inversion task is called reconstruction [10] while the classifier inversion task is called model inversion [12]. We do both in this work. To summarize the goals of model inversion are as follows:

Feature Extractor Given $y = f_T(x)$ find x' that is similar to original x,

Classifier For class i, find x' that is labelled i with high confidence and cannot be distinguished from real image.

4.2. GAN design

Our attack network is a GAN [35]. A GAN architecture has two sub-models, a generator which generates images and a discriminator which judges how good the generated images are. Usually, the input of the generator is a noise vector sampled from a multivariate normal distribution. In our attack case the generator of the inversion attack model is an autoencoder which takes input a feature vector (or a prediction vector) and tries to reconstruct the corresponding image by minimizing multiple loss functions. The core of prior biometrics model inversion attacks is also a GAN [12, 13].

The discriminator and generator of a GAN model can be summarized in two loss equations:

$$L(D) = -E_{x \square P_{real}}[\log(D(x))] - E_{x' \square P_{fake}}[\log(1 - D(x'))] \quad (1)$$

$$L(G) = -E_{x' \square P_{fake}}[\log(D(x'))]. \quad (2)$$

Where $L(D)$ is the discriminator loss and $L(G)$ is the generator loss. x and x' are the original and inverted image. That is, the discriminator's loss function is simply the difference in its classification performance for original and inverted images, while the generator's loss function is how well the discriminator does in identifying fake images.

For feature extractor target networks, the GAN model takes as input a feature vector. For classification networks, the GAN takes as input a classification vector. Recall that these two attacks have different goals, the feature extractor GAN is trying to reproduce x' as accurately as possible. We do not address explicitly train our models to generate x' samples which will have a high inversion attack accuracy. Minimization of visual difference between original and reproduced samples serves as a placeholder for inversion attack accuracy. The generator loss functions include the L1 loss, SSIM [46] loss and the perceptual loss [47] between the inverted and actual image. We minimize: 1) the L1, 2) the perceptual loss, and 3) the structural dissimilarity (or maximizing structural similarity). Finally since a GAN consists of a generator and a discriminator, the generator is fine tuned by the output of the discriminator. Our final objective for the generator including the discriminator loss is:

$$L_G = L_{Perceptual} + L_{L1} + L_{SSIM} + L_D \quad (3)$$

Where L_D is the discriminator output affecting the gener-ator along with other reconstruction loss function as in [13].

4.3. Incorporating multiple vectors

There are multiple ways the additional vectors (for both attack types) per image can be used to better train our attack models. Generally a deep learning feature extraction model is trained by sampling from a training dataset and minimizing or maximizing a loss function.

In the Rand approach, for every update to our attack models we sample vectors randomly (with a probability

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

of $1/\alpha$) chosen from the outputs of one model among α models. The inversion model then learns to invert these feature vectors to their corresponding images.

New mechanisms Merging vectors to form a long vector is another way of feeding additional information to our attack models. The Concat method takes α vectors of size m to form an input vector of size $\alpha \cdot m$. The attack now learns from multiple models in one training step. In the structured random or SR approach, we randomly sample a vector as in our random approach but instead of a m sized vector we form a $\alpha \cdot m$ sized vector with all zeros except the randomly sampled vector placed in the i^{th} index :

$$0, 0, \ldots, 0, f_{T_i}(x_j), 0, \ldots, 0.$$

Intuitively, we force the inversion model to differentiate between vectors gathered from multiple models. This enables the inversion model to learn how the output of a target model changed as it trained (or untrained) to convergence. The attack model now learns from a single vector in a single learning step while having the context of multiple vectors across multiple learning steps.

The structured random w/ alignment loss or SRwAL forces the attack model to predict the i^{th} index or the index which holds the non-zero vector. In the SRwAL method we add to the GAN an additional loss $L_A = \sigma(z)$ where σ represents the softmax function and cross-entropy loss applied on an intermediary layer z in the generator model. This layer predicts the index of the randomly chosen vector. This prediction forces the generator model to implicitly learn features from multiple vectors extracted from multiple models.

This forces the model to further differentiate between vectors from multiple models by forcing the attack model to pass index information across its weights. Alignment loss allows the attack model to better understand how a target model was trained. We show the setup for SRwAL in Figure 1.

4.4. Measuring Success

We use two standard accuracy metrics that will be used in this work for feature extractors [12, 13].

Rank-1 How frequently the inverted biometric value $x^{\vec{}}$ is closest to a biometric from the same class excluding the reading used to invert. A true positive for rank-1 accuracy is when the reconstructed image's extracted feature vector is closest to a feature vector belonging to a member of the same class as the target image. Importantly, it excludes the target image from this comparison.

That is, the true positive rate is for a set of different biometrics $Bio = \{Bio_j\}$ consisting of pairs $x_{i,j}, y_{i,j} = f_T(x_{i,j}) \in Bio_j$:

$$Pr\left[\underset{(x^{\vec{}}, y^{\vec{}}) \in Bio, y^{\vec{}} = y_{i,j}}{\arg\min} d(y^{\vec{}}, f_T(f_T^{-1}(y_{i,j}))) \in Bio_j\right]$$

Type1 Type1 considers the quality of biometric with respect to a specific distance threshold t. That is, we first compute t as the maximum value such that the false accept rate (FAR) of an image of a different biometric (in the underlying f_T) is at most .01 on the target model's training dataset. It then considers how frequently the reconstructed image produces a feature vector that would be accepted by a system with threshold t. Mathematically, this is written as

$$Pr[d(y_{i,j}, f_T(f_T^{-1}(y_{i,j})) \leq t].$$

Rank-1 accuracy is more instructive for applications with all to all matching while Type1 accuracy is more important for a spoofing application where one wishes to break into a biometric authentication system.

For classification networks, we consider traditional accuracy:

Accuracy for y, how frequently is $f_T(f_T^{-1}(y))$ labelled with the same class as y.

In all attacks, we do not use any images used to train the target model in the attack. Instead, we probe the target model with a probe dataset that is smaller than the training dataset. This probe dataset is class disjoint from the training dataset for the feature extraction setting.

5. Evaluation

This section details the datasets used in training both the target models and the attack, specifies the training methodology for the target models, and describes accuracy metrics used for the attacks.

We utilize two datasets in our work, one for the iris and one for faces. The ND-IRIS-0405 [48, 49] dataset contains 64,980 iris grayscale images from 356 subjects. The classes are highly imbalanced, some classes have many more images than others. There are 712 classes since left and right irises are different classes due to them being statistically independent [50].

Labeled Faces in the Wild (LFW) [51] face recognition dataset contains 13233 images of 5749 people downloaded from the various websites with 1680 people having 2 or more images. For our evaluation we only consider people with more than 15 images yielding 89 classes or people with 3482 images.

5.1. Target models training

Our target models use the DenseNet-169 architecture from the original DenseNet paper [52]. We use loss function from SphereFace [6] coupled with the Adam optimizer [53] to train the target networks using Tensorflow [54]. Dropout [55] has been studied in literature as a defense against membership inference attacks [56]. We train our

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

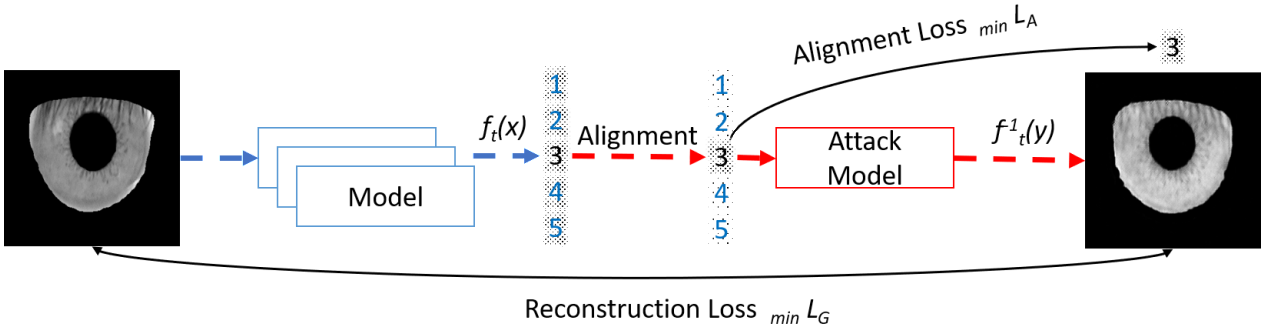

Figure 1: Vector alignment process for SRwAL method of incorporating multiple vectors. When reconstructing from vectors only a single feature vector is used while the rest are truncated to zero. The inversion model now implicitly learns from multiple vectors over the entire training process.

networks with dropout applied to the fully connected layer which is the second to last layer of our target classification network. The dropout ratio used is 0.5. DenseNets provide near state of art recognition accuracy when coupled with dropout. Our target networks are thus generalized and possess some defense through the use of Dropout. Mai et al. [12] and Ahmad et al. [13] do not use any dropout in their target networks.

We now discuss the training and probe dataset splits for our feature extraction and classification networks. Note that in our attacks against feature extraction networks the target data's training is class disjoint from the probe attack dataset. This is not the case for classification networks as classification networks are designed for a predefined set of classes.

Iris - Feature Extraction The target model for the iris dataset is trained on left iris images of all (356) subjects forming a private training set of roughly 10000 images. The attack model is trained on a dataset of 2000 right iris images randomly sampled from all classes. The testing set (probe) for the attack model also has 2000 images from left irises of all subjects. This probe dataset is disjoint (but not class disjoint) from the private training dataset used to train the target model.

Iris - Classification We train our target model on left iris images of all subjects. The total number of images is 11000 with training done on 7000. 2000 left iris images from these subjects are used to train the attack model and the remaining 2000 are used as testing for the attack model .

Face - Feature Extraction After restricting to faces with at least 15 images there are 89 classes with 3632 total images. Of these 1076 are used to train the target network, 1500 are used to train the attack model while the remaining 1556 are used as probe images as testing for the attack model.

Face - Classification The target model is trained on all entire 89 classes leaving out 15% images from each class to make the probe dataset.

The iris images are segmented [57] to not include any additional texture besides that of the iris. The reconstruction attack model therefore is forced to learn texture information stored in the output feature vector. We utilize deep-funneled images [58] for LFW dataset and crop the images to a size of 128x128 to include the face area only.

Dataset	Types of Models	#	Training set size	f_E Type1	f_E Rank-1	f_C Acc.
ND	Single	1	2000	59%	35%	81%
	Upslope	5	2000	65%	45%	82%
	Update	3	2000	61%	38%	81%
	Downslope	5	2000	60%	44%	82%
	[13]	1	20000	75%	96%	-
LFW	Single	1	1500	85%	82%	74%
	Upslope	5	1500	89%	84%	78%
	Update	3	1500	87%	84%	75%
	Downslope	5	1500	87%	83%	73%
	[12]	1	$2 \cdot 10^6$	99%	-	-

Table 1: Comparison of Accuracy when using multiple models with the Rand method of incorporation. Model Inversion for both Feature Extraction Networks, f_E and Classification Networks f_C. Accuracy per dataset and attack type. Accuracy for classification networks is how frequently an image is assigned the correct class label.

6. Results

6.1. Feature Extraction Networks

We show Type-1 and Rank-1 accuracy for our attacks. An overview of results in Tables 1 and 2. Figure 3 showed visual results for the iris. Visual results for the facial biometric are in Figure 2.

Single Model Results Type-1 attack accuracy when inverting feature vectors using access to a single target model is 59% and 85% for the iris and face dataset respectively. In the Type-1 setting a reconstructed biometric is matched with its original counterpart, we obtain Type-1 attack accuracy numbers of 59% compared to Ahmad and Fuller [13] who achieve 75% while using 10 times the training set size. These results are shown in Table 1.

Our Rank-1 accuracy is considerably lower . We attribute this to slight overfitting of the inversion network due to our small training dataset and the use of dropout in the target network. Our target network also uses a more modern loss function than Ahmad and Fuller [13]. Rank-1 accuracy measures the probability of a reconstructed biometric being matched with an original biometric of the same class (and not itself), our inversion network seems to do better at the specific task of inverting a template to a particular image and

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

Dataset	Incorporation Method	# Models	Training set size	Models for Test	f_E Type1	f_E Rank-1	f_C Accuracy
ND	Rand	5	2000	Final	65%	45%	82%
	Concat	5	2000	Final	48%	27%	78%
	Concat	5	2000	All	50%	30%	86%
	SR	5	2000	Final	66%	46%	81%
	SRwAL	5	2000	Final	72%	53%	83%
	SRwAL	5	2000	All	65%	45%	81%
	[13]	1	20000	–	75%	96%	-
LFW	Rand	5	1500	Final	89%	84%	78%
	Concat	5	1500	Final	78%	75%	74%
	Concat	5	1500	All	80%	78%	81%
	SR	5	1500	Final	89%	84%	79%
	SRwAL	5	1500	Final	91%	86%	79%
	SRwAL	5	1500	All	89%	84%	79%
	[12]	1	2000000	–	99%	-	-

Table 2: Comparison of Methods for incorporating multiple models. All data uses Upslope models. Both Feature Extraction Networks, f_E and Classification Networks f_C. Accuracy per dataset and attack type. Accuracy for classification networks is how frequently an image is assigned the correct class label. Models for Test Column indicates whether all models or just the final model were used during testing.

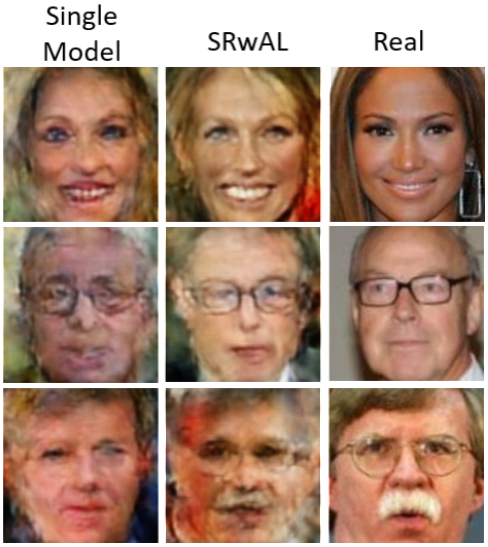

Figure 2: Alignment helps with reconstructing face features such as facial hair and correcting skin tone.

does not generalize well.

Our target network has also not been fine tuned, our Rank-1 accuracy on the test set of the target network is 98.2% while Ahmad and Fuller used a target network with an accuracy of 99.5%. This accuracy changes the threshold distance used to accept or reject biometric comparisons; this change affects Type1 attack accuracy but not Rank-1.

For LFW, we achieve a Type-1 accuracy of 85% when using a single model to obtain the training dataset for the inversion attack network. Our inversion attack network achieves 85% accuracy in Type-1 and 82% in Rank-1 settings. This is in contrast to iris reconstruction results, face images have myriad of facial features in addition to some background of the LFW images making them easier to invert and harder for the target model to achieve high test accuracy. Iris images have only the iris texture while other features such as the skin and eyelid are segmented out.

Incorporating Multiple Models Turning to the setting of multiple models, we present the gain in using multiple models with the Rand technique in Table 1. In all settings, multiple models improve accuracy of the inversion network. Because all attack settings perform similarly, in comparing how to effectively incorporate multiple models we focus on the Upslope model.

Results for different incorporation techniques are presented in Table 2. The largest gain is using the technique SRwAL. For the iris, this technique boosts Rank-1 accuracy from 45% to 53%. Input-augmented GANs boost attack accuracy in most settings.

The SR technique performs nearly identically to the Rand technique. This is of particular interested compared to the SRwAL technique which is only forcing the model to learn the provided input which is easy.

Discussion The Concat technique can hurt performance when only a single model output is available at testing. The most natural explanation is that feature vectors from models which have not converged hold information that is hard to use by our inversion models. However, if one assumes that the adversary sees the output of all models at test time, that is the adversary sees $f_{T_1}(x), \ldots, f_{T_\alpha}(x)$ at test time this accuracy improves. This indicates that the problem may be the mismatch between the format of the training and testing data. We note that this phenomenon is switched on SRwAL, providing all vectors at test time is harmful. This supports the hypothesis that Structured Random with Alignment loss is superior for natural attack scenarios.

The accuracy gain when using multiple models is less

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

pronounced for the LFW face dataset. In this setting, we believe that the small amount of training data resulting in the attack model overfitting the training data. However, our attack achieves close to state of art inversion accuracy while using orders of magnitude less training data.

6.2. Classification Networks

Model inversion attacks on classification networks output the average of a certain class (see discussion in Section 4.1). The attack is successful if the reconstructed biometric images are classified to their correct class by the target model. For Single Model Results our inversion attack models perform at 81% and 74% attack accuracy for the iris and face datasets respectively. Results are displayed in Table 1.

For Incorporating Multiple Models structured random with alignment loss bumps the accuracy to 83% and 79% respectively. We do not see a proportional increase in inversion accuracy as we saw with feature extraction networks. Classification networks output prediction vectors which are simple and do not hold much information. Previous works have even truncated prediction vectors [36] for better inversion.

If all models' output is available at test time Concat method improves but the SRwAL does not. This same phenomenon was observed in feature extraction network.

Recall, for classification networks the traditional goal is to output the class average. Prior works have not considered that this average may not appear similar to a real biometric (such as Fredrikson et al. [3]). When training and testing with concatenated prediction vectors ($\alpha \cdot m$) our inverted images vary across a class instead of being the same class average image. An example of different images for the same iris biometric is shown in Figure 3. We attribute this to the additional information in multiple vectors which form the concatenated vector. A similar phenomenon is seen in the work of Yang et al. [36] where classes unknown to the target model are inverted by a method called alignment (that differs from SRwAL).

6.3. Which attack types perform the best?

We perform an experiment to validate which models contribute the most to inversion attack accuracy. With access to the final trained target model and using random sampling for training, an adversary's reconstructed iris images have a Rank-1 attack accuracy of 35% which drops to 15% if the adversary has access to the 25th and 50th model. Attack accuracy jumps to 36% if the first and 100th model are used by the adversary. The target model is trained using an off the shelf network architecture which was pre-trained on the Imagenet dataset, which would generate different yet somewhat accurate feature vectors [59]. Finally the last model would generate accurate feature vectors which would allow the adversary to generate better reconstructions. Our proposed Alignment loss works best with feature vectors while for prediction vectors the concatenation attack works best.

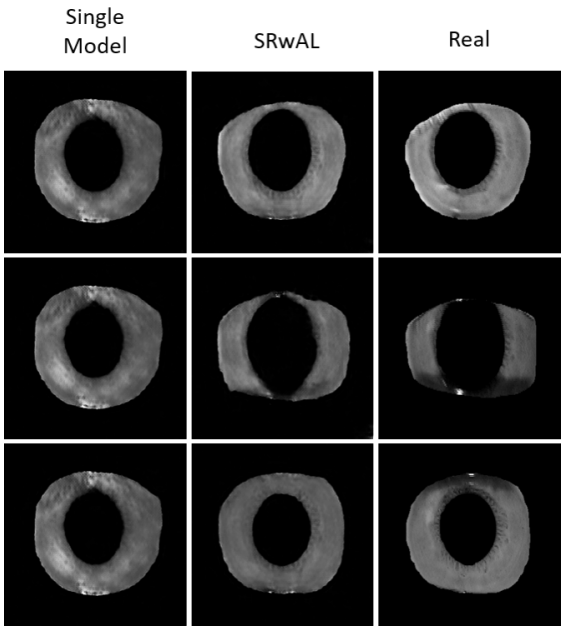

Figure 3: Alignment process enables inversion to vary. Each row represents a different iris from the same biometrics. For classification networks single model always inverts to class average. However, SRwAL can invert to distinct images that better match the stored template.

7. Conclusion

An adversary can perform model inversion attacks to gain unauthorized access to biometric authentication systems through biometric spoofing. We explore an adversary's access to deep learning models trained and stored, models generated after a model is updated, and finally models generated after an unlearning request. In this work we show when multiple models are accessible by an adversary model inversion attacks can be performed with fewer training samples with high attack accuracy. We explore different methods of incorporating multiple models into the attack model training process.

An interesting finding of our work is that while incorporating multiple models using the Rand method is universally helpful (across biometrics and types of biometric transforms), results using input-augmented GANs are mixed. If only the last model is available at Test time the Concat technique can actually hurt performance, for the iris Type1 accuracy drops from 59% to 48% and is much lower than the 65% achieved by the random method. However, our proposed method of using SRwAL always improves performance compared to the Rand technique improving Type1 accuracy to 72% compared to the 65% of Rand. We leave additional measures of combining data from multiple models and performance with additional data as future work.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

References

[1] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," IBM systems Journal, vol. 40, no. 3, pp. 614–634, 2001.

[2] A. K. Jain, P. Flynn, and A. A. Ross, Handbook of biometrics. Springer Science & Business Media, 2007.

[3] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 1322–1333.

[4] S. Marcel, M. S. Nixon, and S. Z. Li, Handbook of biometric anti-spoofing. Springer, 2014, vol. 1.

[5] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4690–4699.

[6] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 212–220.

[7] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5265–5274.

[8] K. Wang and A. Kumar, "Cross-spectral iris recognition using cnn and supervised discrete hashing," Pattern Recognition, vol. 86, pp. 85–98, 2019.

[9] Z. Zhao and A. Kumar, "Towards more accurate iris recognition using deeply learned spatially corresponding features," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3809–3818.

[10] S. Ahmad and B. Fuller, "Thirdeye: Triplet-based iris recognition without normalization," in IEEE International Conference on Biometrics: Theory, Applications and Systems, 2019.

[11] 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 253–261.

[12] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain, "On the reconstruction of face images from deep face templates," IEEE transactions on pattern analysis and machine intelligence, vol. 41, no. 5, pp. 1188–1202, 2018.

[13] S. Ahmad and B. Fuller, "Resist: Reconstruction of irises from templates," in International Joint Conference on Biometrics, 2020.

[14] A. Salem, A. Bhattacharya, M. Backes, M. Fritz, and Y. Zhang, "Updates-leak: Data set inference and reconstruction attacks in online learning," in 29th USENIX Security Symposium, 2020, pp. 1291–1308.

[15] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in 25th USENIX Security Symposium), 2016, pp. 601–618.

[16] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018, pp. 36–52.

[17] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4954–4963.

[18] J. Zuo, N. K. Ratha, and J. H. Connell, "Cancelable iris biometric," in 2008 19th International Conference on Pattern Recognition. IEEE, 2008, pp. 1–4.

[19] M. Gomez-Barrero, C. Rathgeb, J. Galbally, C. Busch, and J. Fierrez, "Unlinkable and irreversible biometric template protection based on bloom filters," Information Sciences, vol. 370, pp. 18–32, 2016.

[20] J. Bringer, C. Morel, and C. Rathgeb, "Security analysis of bloom filter-based iris biometric template protection," in 2015 international conference on biometrics (ICB). IEEE, 2015, pp. 527–534.

[21] M. Stokkenes, R. Ramachandra, M. K. Sigaard, K. Raja, M. Gomez-Barrero, and C. Busch, "Multi-biometric template protection—a security analysis of binarized statistical features for bloom filters on smartphones," in 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA). IEEE, 2016, pp. 1–6.

[22] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," SIAM journal on computing, vol. 38, no. 1, pp. 97–139, 2008.

[23] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in Proceedings of the 6th ACM conference

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

on Computer and communications security, 1999, pp. 28–36.

[24] A. Juels and M. Sudan, "A fuzzy vault scheme," Designs, Codes and Cryptography, vol. 38, no. 2, pp. 237–257, 2006.

[25] Z. Jin, J. Y. Hwang, Y.-L. Lai, S. Kim, and A. B. J. Teoh, "Ranking-based locality sensitive hashing-enabled cancelable biometrics: Index-of-max hashing," IEEE Transactions on Information Forensics and Security, vol. 13, no. 2, pp. 393–407, 2017.

[26] X. Boyen, "Reusable cryptographic fuzzy extractors," in Proceedings of the 11th ACM conference on Computer and Communications Security, 2004, pp. 82–91.

[27] B. Fuller, X. Meng, and L. Reyzin, "Computational fuzzy extractors," in International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2013, pp. 174–193.

[28] F. Hernández Álvarez, L. Hernández Encinas, and C. Sánchez Ávila, "Biometric fuzzy extractor scheme for iris templates," 2009.

[29] J. Bringer, H. Chabanne, G. Cohen, B. Kindarji, and G. Zémor, "Optimal iris fuzzy sketches," in 2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems. IEEE, 2007, pp. 1–6.

[30] D. Keller, M. Osadchy, and O. Dunkelman, "Fuzzy commitments offer insufficient protection to biometric templates produced by deep learning," arXiv preprint arXiv:2012.13293, 2020.

[31] R. Canetti, B. Fuller, O. Paneth, L. Reyzin, and A. Smith, "Reusable fuzzy extractors for low-entropy distributions," Journal of Cryptology, vol. 34, no. 1, pp. 1–33, 2021.

[32] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017, pp. 19–38.

[33] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 13, no. 3, pp. 1–207, 2019.

[34] B. D. Rouhani, M. S. Riazi, and F. Koushanfar, "Deepsecure: Scalable provably-secure deep learning," in Proceedings of the 55th Annual Design Automation Conference, 2018, pp. 1–6.

[35] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.

[36] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via background knowledge alignment," in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 225–240.

[37] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan et al., "Towards federated learning at scale: System design," Proceedings of Machine Learning and Systems, vol. 1, pp. 374–388, 2019.

[38] 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 23rd {USENIX} Security Symposium ({USENIX} Security 14), 2014, pp. 17–32.

[39] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in 2015 IEEE Symposium on Security and Privacy. IEEE, 2015, pp. 463–480.

[40] L. Bourtoule, V. Chandrasekaran, C. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," arXiv preprint arXiv:1912.03817, 2019.

[41] A. Ginart, M. Guan, G. Valiant, and J. Y. Zou, "Making AI forget you: Data deletion in machine learning," in Advances in Neural Information Processing Systems, 2019, pp. 3518–3531.

[42] A. Mantelero, "The eu proposal for a general data protection regulation and the roots of the 'right to be forgotten'," Computer Law & Security Review, vol. 29, no. 3, pp. 229–235, 2013.

[43] N. F. Palmieri III, "Who should regulate data: An analysis of the california consumer privacy act and its effects on nationwide data protection laws," Hastings Sci. & Tech. LJ, vol. 11, p. 37, 2020.

[44] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.

[45] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017, pp. 3–18.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.

[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE transactions on image processing, vol. 13, no. 4, pp. 600–612, 2004.

[47] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in European conference on computer vision. Springer, 2016, pp. 694–711.

[48] K. W. Bowyer and P. J. Flynn, "The ND-IRIS-0405 iris image dataset," arXiv preprint arXiv:1606.04853, 2016.

[49] P. J. Phillips, K. W. Bowyer, P. J. Flynn, X. Liu, and W. T. Scruggs, "The iris challenge evaluation 2005," in Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on. IEEE, 2008, pp. 1–8.

[50] J. Daugman, "Iris recognition border-crossing system in the uae," International Airport Review, vol. 8, no. 2, 2004.

[51] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[52] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.

[54] M. Abadi, "Tensorflow: learning functions at scale," in Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, 2016, pp. 1–1.

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.

[56] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," arXiv preprint arXiv:1806.01246, 2018.

[57] S. Ahmad and B. Fuller, "Unconstrained iris segmentation using convolutional neural networks," in Asian Conference on Computer Vision. Springer, 2018, pp. 450–466.

[58] G. B. Huang, M. Mattar, H. Lee, and E. Learned-Miller, "Learning to align from scratch," in NIPS, 2012.

[59] A. Boyd, A. Czajka, and K. Bowyer, "Deep learning-based feature extraction in iris recognition: Use existing models, fine-tune or train from scratch?" in 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS). IEEE, 2019, pp. 1–9.

Authorized licensed use limited to: UNIVERSITY OF CONNECTICUT. Downloaded on June 07,2023 at 16:38:55 UTC from IEEE Xplore. Restrictions apply.