Model Inversion Attack against a Face Recognition System in a Black-Box Setting

Shunsuke Yoshimura*, Kazuaki Nakamura*, Naoko Nitta*, and Noboru Babaguchi*
* Graduate School of Engineering, Osaka University, Osaka, Japan
E-mail: {yoshimura, k-nakamura, naoko, babaguchi}@nanase.comm.eng.osaka-u.ac.jp Tel: +81-6-6879-7746

***Abstract*—A DNN-based face recognition system implicitly has the information of facial characteristics of the individuals registered in it. The information could be maliciously revealed or stolen by a model inversion attack (MIA), which causes a serious privacy issue. To clarify how much the threat of MIA is real, methods to perform MIA against a face recognition system have been studied in recent years. Theoretically, MIA is formulated as a problem of finding the best image that maximizes the recognition score outputted by a target recognition system. This can be achieved by a gradient descent technique if the target system is a white box whose network structure and parameters are known, as assumed in the most existing methods. However, this assumption is not necessarily realistic. Unlike the existing methods, in this paper, we propose an MIA method that can be carried out against a black-box system. To enable the proposed method to generate natural-looking face images, we first introduce a deep face generator that generates a face image from a random feature vector, by which MIA is re-defined as a problem of finding the best feature vector instead of the best image. The proposed method solve this problem by a gradient descent technique, where we numerically approximates the gradient of the recognition score by perturbing the current feature vector several times. Our experimental results demonstrate that the proposed method can generate natural-looking face images successfully containing personal facial characteristics, whose performance is comparable to the white-box-oriented existing methods.**

I. Introduction

The performance of image recognition systems has rapidly increased in recent years and is still growing now with the progress of deep neural networks (DNN). The state-of-the-art recognition methods are actively developed into cloud services, whose typical examples include Google Cloud Vision [1], Amazon Rekognition Image [2], and so on. On the other hand, attacks on deep learning-based image recognition systems have also been explored. What kinds of attacks could be done by malicious people? How much are the attacks real and serious? These questions should be carefully discussed to make the cloud-based image recognition services safer and more secure.

Model inversion attack, abbreviated as MIA [3], [4], is one of the attacks attracting many researchers' attention in the field of multimedia security. Given a certain class label arbitrarily specified by an attacker, the MIA process aims to generate a natural-looking image that is recognized as the given class by a target recognition system. This could cause a serious privacy issue. For instance, if the target is a face recognition/identification system in which many individuals registered, their face images could be generated only from a class label (i.e., their name) by MIA, which might be unauthorizedly distributed and misused for impersonation or fake media generation. To measure the risk of MIA and develop its countermeasures, investigating possible methods of MIA is very urgent.

So far, several existing studies have proposed an MIA method that can generate highly natural images under the white-box setting, which means the network structure and the parameters of the target system are known [5]. Although the knowledge provided by these studies is important, their assumed white-box scenario is not so realistic; most of the actual cloud services of image recognition do not open such information to the public.

Unlike the existing studies, in this paper, we propose an MIA method that works well in the black-box setting, which means the network structures and the parameters of the target system are not disclosed and therefore the attacker cannot use them. We particularly focus on a face recognition system as the target of MIA and introduce a deep face generator that generates a face image from a random feature vector. The goal of the proposed method is to find the best feature vector whose corresponding face image is natural as well as recognizable as the individual arbitrarily specified by the attacker. To this end, we employ the gradient descent-based optimization strategy, but we do not compute the theoretical value of the gradient vector at each iteration of the gradient descent algorithm. Instead, the proposed method numerically computes its approximation by adding perturbations to the current feature vector. The contributions of this research work are summarized as follows.

- This paper proposes an MIA method working well in the black-box setting, which demonstrates that MIA can be performed in a realistic situation.
- Once the deep generator is pre-trained, the proposed method does not need any training process for MIA, which is much efficient than existing methods.
- We also provide a criterion to decide the appropriate size of the perturbations for the numerical gradient approximation, which further improves the effectiveness and the efficiency of the proposed method.

The remainder of this paper is organized as below. First, we review some related studies in Section II. Next, we explicitly describe the problem setting assumed in most MIA scenarios in Section III before providing the details of the proposed

method in Section IV. Then we experimentally evaluated the proposed method in Section V and finally conclude this paper in Section VI.

II. Related Work

Attacks on image recognition systems have been discussed in literature since more than a dozen years ago before DNNs have become mainstream in this area. Huang et al. categorized the possible attacks into the following two categories: causative attacks (CA) and exploratory attacks (EA) [6]. The former degrades the performance of a target system by directly altering its recognition model or training dataset. Intentionally sending some outliers to a target system that continually updates its model by an online algorithm is a typical example of CA [7], [8]. On the other hand, the latter, EA, analyzes the recognition model of the target system to steal or reveal some private information contained in the model. The attack of generating adversarial examples is a kind of EA, which can be regarded as a plot of revealing the weak points of the target system [9], [10]. Model extraction attack, which aims to steal the target model itself and unauthorizedly make its duplication, is another example of EA [11], [12], [13], [14], [15], [16]. The focus of this paper, namely MIA, is also categorized into EA.

MIA first appeared in Fredrikson's work in 2014 [3], where the authors considered that input data to a pattern recognition system consists of privacy-sensitive parts and non-sensitive parts. For instance, in the case of face images, the detailed shape of the eyes and that of the mouth are privacy-sensitive while the rough contour of the whole head region is non-sensitive. Based on the above consideration, Fredrikson et al. defined MIA as a process of estimating the sensitive parts from the non-sensitive parts in addition to a given class label. They applied their MIA method to a linear regression model, decision trees, and shallow neural networks in their work [3], [4] and reported interesting results. Fredrikson's work was followed by Wu et al. in 2016 [17], where MIA was formulated more theoretically. Recently, not only pattern recognition systems but also recommendation systems have been considered as a target of MIA [18], [19].

There are two types of MIA methods against image recognition systems: training-based and optimization-based. In the former, an inverse model of the target system is tried to be directly constructed. Theoretically, an image recognition system is regarded as a map from the image domain to the score vector domain, where each element in the score vector represents the probability of the input image to be recognized as the corresponding class. Hence, an inverse model of the target system, that is, a map from the score vector domain to the image domain can be designed, which allows the attackers to perform MIA. Yang et al. employed this approach [20]. However, since the dimension of the image domain is much higher than that of the score vector domain, the target system generally forms a many-to-one map. Thus, its inverse model becomes a one-to-many map, which is difficult to stably train.

On the other hand, the latter, namely an optimization-based approach, tries to find the best input image that minimizes a certain loss function. Fredrikson's original method is categorized into this type. Zhang et al. also employed this approach [5]. To generate natural-looking images by MIA, they utilized a generative adversarial network (GAN) [21], which was trained by minimizing a loss function consisting of the adversarial loss and the identity loss. The identity loss becomes large if the generated image is incorrectly recognized by the target recognition system. Although this method achieves good performance, it is computationally heavy because a GAN has to be separately trained for each class. Moreover, it does not work in the black-box setting because the network structure and the parameters of the target system should be known to perform a gradient descent algorithm in the training process of the GANs.

In this paper, we also employ the optimization-based approach and aim to propose an efficient MIA method that works well in the black-box setting.

III. Assumptions and Basic Strategy of MIA

In this section, we specify which kinds of information can be obtained and how they are exploited by the attackers in the common scenario of MIA that is employed in most existing studies.

A. Assumptions on the Target System

The target system R is designed as a neural network that receives a face image $x \in X$ as an input and outputs a score vector $\boldsymbol{y} = (y_1 \ \cdots \ y_d)^\top = R(x) \in \mathbb{R}^d$, where X is a set of all possible images and d is the number of individuals registered in the R. The n-th element of the score vector, y_n, indicates how much likely the input image x is the n-th individual's face, where $0 \leq y_n \leq 1$. In other words, $y_n = p(n|x)$ is the x's probability of being recognized as the n-th class. The owner of the R collects her own image dataset T to train it. Since the T might contain some private information, it is not disclosed to the public; only the R itself is open to its users. When the users send an image x to the R, it computes $\boldsymbol{y} = R(x)$ and returns it to the users as the recognition result. In practice, the $\boldsymbol{y}$ might be partially masked before returned to the users (e.g., only five elements having the highest scores are returned), but in the common MIA scenario, all the elements in $\boldsymbol{y}$ are returned for any $x \in X$.

B. Assumptions on the Attackers

From the viewpoint of the MIA attackers, they cannot access nor exploit T that is owned by the R's owner. However, they are allowed to collect another image set S that is independent of T. The attackers can get all the elements in $\boldsymbol{y} = R(x)$ for any $x \in S$ when they send it to the R. There is no strict limitation on the number of uses of the R. In the white-box setting, the network structure of the R and its parameters are disclosed and could be exploited by the attackers. On the other hand, in the black-box setting, the attackers do not know such information.

C. Basic Strategy of MIA

Suppose the case that the attacker wants to generate the n-th individual's face image. To this end, he first sets a one-hot vector $\hat{\boldsymbol{y}}$, whose n-th element is 1 and all the other elements are 0. Then he tries to generate $\hat{x} \in X$ that satisfies $R(\hat{x}) = \hat{\boldsymbol{y}}$. This can be obtained as

$$\hat{x} \;=\; \underset{x}{\operatorname{argmin}}\{L(\boldsymbol{y};\hat{\boldsymbol{y}})\} \;=\; \underset{x}{\operatorname{argmin}}\{L(R(x);\hat{\boldsymbol{y}})\}\;, \qquad (1)$$

where L is a certain loss function. Any kind of functions such as the mean squared error and the cross entropy can be used as L, as long as its minimum value is obtained when and only when $\boldsymbol{y} = \hat{\boldsymbol{y}}$. In this paper, we employ the cross entropy loss because of the following reason.

In terms of the probability theory, the goal of the attacker is to generate $\hat{x}$ that maximizes $p(x|n)$, where $p(x|n)$ is the probability density of the n-th individual's face images over X. Using the Bayes' theorem, we can derive

$$p(x|n) \;=\; \lambda p(x)p(n|x) \;=\; \lambda p(x)y_n \;=\; \lambda\eta y_n \qquad (2)$$

with the general assumption of the uniform prior $p(x) = \eta$, where $\lambda = \frac{1}{p(n)}$ is a positive constant. Due to the monotonicity of the logarithm function, maximization of y_n is equivalent with minimization of $-\log y_n$. In addition, we can also derive

$$-\log y_n \;=\; -\sum_{i=1}^{d} \hat{y}_i \log y_i \;=\; \mathrm{CrossEntropy}(\boldsymbol{y};\hat{\boldsymbol{y}}) \qquad (3)$$

using the above $\hat{\boldsymbol{y}}$, where $\hat{y}_i$ is the i-th element of the $\hat{\boldsymbol{y}}$. Hence, the attacker can obtain $\hat{x}$ as

$$\begin{aligned}\hat{x} \;&=\; \underset{x}{\operatorname{argmax}}\{p(x|n)\}\\ &=\; \underset{x}{\operatorname{argmin}}\{-\log y_n\}\\ &=\; \underset{x}{\operatorname{argmin}}\{\mathrm{CrossEntropy}(\boldsymbol{y};\hat{\boldsymbol{y}})\}\;,\end{aligned} \qquad (4)$$

which becomes equivalent with Eq. (1) by using the cross entropy loss as the L.

IV. MIA Method in a Black-Box Setting

In this section, we describe the proposed MIA method in detail. To make the description straightforward, we start from the case of the white-box setting in Subsection IV-A and then move to the case of the black-box setting in the subsequent subsections.

A. MIA by Gradient Descent with a Deep Generator

Based on the discussions in Section III, our focus is to minimize the loss function $L(R(x);\hat{\boldsymbol{y}})$ introduced in Eq. (1). Gradient descent is a straightforward way to solve this problem. Starting from an initial seed image $x^{(0)}$, we iteratively update it as

$$\begin{aligned}x^{(t+1)} \;&=\; x^{(t)} - \alpha\,\frac{\partial L}{\partial x}\big(x^{(t)}\big)\\ &=\; x^{(t)} - \alpha\,\frac{\partial L}{\partial \boldsymbol{y}}\big(\boldsymbol{y}^{(t)}\big)\frac{\partial R}{\partial x}\big(x^{(t)}\big)\end{aligned} \qquad (5)$$

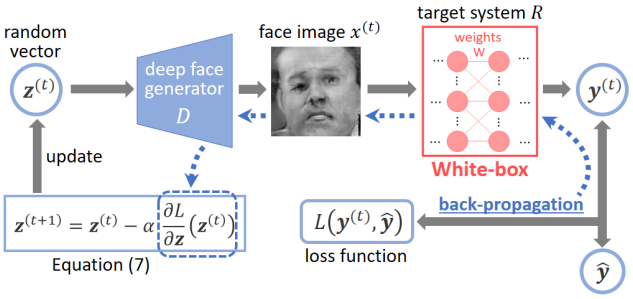

Fig. 1. MIA method with deep generator (in white-box setting).

where $\boldsymbol{y}^{(t)} = R(x^{(t)})$ and $t = 0,\ 1,\ 2,\ \cdots$. The positive constant α is called "learning-rate", which controls the balance between the speed and stability of convergence. Theoretically, we can obtain $\hat{x}$ by repeating the above updating process enough times. However, the $\hat{x}$ obtained by this approach is often similar to its seed $x^{(0)}$; in other words, $\hat{x}$ behaves as a kind of adversarial example for the target system R. Indeed, the above approach is similar to a standard generation process of adversarial examples. This problem is caused because the value of each pixel in $x^{(t)}$ is independently updated from all the other pixels, which can be avoided by introducing a deep face generator D.

In the proposed method, the generator D is pre-trained so that it can generate a natural-looking face image from a m-dimensional random feature vector $\boldsymbol{z} \in \mathbb{R}^m$. The pre-training process is done only once by the attacker based on his own dataset S. Then $x = D(\boldsymbol{z}) \in X$ is used as an input to the target system, where our focus moves to finding the best $\hat{\boldsymbol{z}}$ that minimizes the loss function $L(R(x);\hat{\boldsymbol{y}}) = L(R(D(\boldsymbol{z}));\hat{\boldsymbol{y}})$, that is,

$$\hat{\boldsymbol{z}} \;=\; \underset{\boldsymbol{z}}{\operatorname{argmin}}\{L(R(D(\boldsymbol{z}));\hat{\boldsymbol{y}})\}\;. \qquad (6)$$

As shown in Fig. 1, this is solved by the gradient descent that updates an initial seed vector $\boldsymbol{z}^{(0)}$ as

$$\boldsymbol{z}^{(t+1)} \;=\; \boldsymbol{z}^{(t)} - \alpha\,\frac{\partial L}{\partial \boldsymbol{z}}\big(\boldsymbol{z}^{(t)}\big) \quad (t = 0,\ 1,\ \cdots)\;, \qquad (7)$$

where

$$\frac{\partial L}{\partial \boldsymbol{z}}\big(\boldsymbol{z}^{(t)}\big) \;=\; \frac{\partial L}{\partial \boldsymbol{y}}\big(\boldsymbol{y}^{(t)}\big)\frac{\partial R}{\partial x}\big(x^{(t)}\big)\frac{\partial D}{\partial \boldsymbol{z}}\big(\boldsymbol{z}^{(t)}\big)\;. \qquad (8)$$

Note that $x^{(t)} = D\big(\boldsymbol{z}^{(t)}\big)$ and $\boldsymbol{y}^{(t)} = R\big(x^{(t)}\big)$.

Since the network structure of the R and its parameters are disclosed in the white-box setting, the attacker can obtain $\frac{\partial R}{\partial x}(x^{(t)})$ by the back-propagation technique at each iteration t. Repeating the above updating process enough times, the attacker finally obtain $\hat{\boldsymbol{z}}$ and compute the resultant image of MIA as $\hat{x} = D(\hat{\boldsymbol{z}})$. As mentioned in Section III, we employ the cross entropy loss as L and therefore $L(R(D(\boldsymbol{z}));\hat{\boldsymbol{y}}) = -\log p(n|D(\boldsymbol{z}))$. Based on this fact, we stop the above iterative process when $\exp\big(-L(R(D(\boldsymbol{z}^{(t)}));\hat{\boldsymbol{y}})\big) > 0.99$ is satisfied in our experiments.

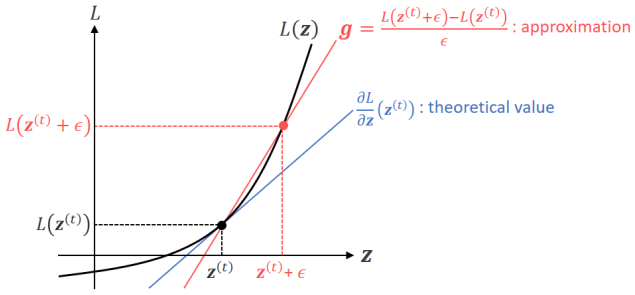

Fig. 2. Perturbation-baased gradient approximiation (in case of one-variable function).

B. Perturbation-based Gradient Vector Approximation

In the black-box setting, the attacker cannot conduct back-propagation to obtain $\frac{\partial R}{\partial x}(x^{(t)})$ due to the lack of information on the structure and parameters of R. Hence, he has to compute $\frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)})$ by another way. Even in this case, $\boldsymbol{y} = R(D(\boldsymbol{z}))$ can be obtained and therefore $L(\boldsymbol{y}; \hat{\boldsymbol{y}}) = L(R(D(\boldsymbol{z})); \hat{\boldsymbol{y}})$ can be computed for any $\boldsymbol{z} \in \mathbb{R}^m$. Hereafter, we regard the L as a function of $\boldsymbol{z}$ and simply write it as $L(\boldsymbol{z})$. To obtain the gradient vector $\frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)})$, we propose to use a perturbed vector $\boldsymbol{z}^{(t)} + \boldsymbol{\epsilon}$ and its corresponding output by L, i.e., $L(\boldsymbol{z}^{(t)} + \boldsymbol{\epsilon})$.

For simplicity of consideration, suppose that $\boldsymbol{z}$ is a scalar, i.e., $m = 1$. In this case, the theoretical value of the gradient $\frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)})$ can be approximated by

$$g = \frac{L(\boldsymbol{z}^{(t)} + \epsilon) - L(\boldsymbol{z}^{(t)})}{(\boldsymbol{z}^{(t)} + \epsilon) - \boldsymbol{z}^{(t)}} = \frac{L(\boldsymbol{z}^{(t)} + \epsilon) - L(\boldsymbol{z}^{(t)})}{\epsilon} \quad (9)$$

using the perturbation ϵ, as shown in Fig. 2. Smaller ϵ leads to a better approximation since the limit of g as ϵ approaches zero is exactly the theoretical value. We extend this idea to the case of multi-dimensional $\boldsymbol{z}$, where perturbing $\boldsymbol{z}^{(t)}$ only once is not enough, thus we perturb it M times.

Let $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \cdots, \boldsymbol{\epsilon}_M$ be the perturbations. Each $\boldsymbol{\epsilon}_k$ is drawn from a normal distribution $\mathcal{N}(0, \sigma^2 I_m)$, where I_m is the m-dimensional identity matrix and σ^2 is the variance of each dimension. For any k, $L(\boldsymbol{z}^{(t)} + \boldsymbol{\epsilon}_k)$ can be expressed in the form of Taylor series, i.e.,

$$L(\boldsymbol{z}^{(t)} + \boldsymbol{\epsilon}_k) = L(\boldsymbol{z}^{(t)}) + \boldsymbol{\epsilon}_k^\top \frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)}) + Q(\boldsymbol{\epsilon}_k) , \quad (10)$$

where $Q(\boldsymbol{\epsilon}_k)$ includes only quadratic and higher degree terms. When $\boldsymbol{\epsilon}_k$ is enough small, $Q(\boldsymbol{\epsilon}_k)$ is approximately zero and

$$L(\boldsymbol{z}^{(t)} + \boldsymbol{\epsilon}_k) - L(\boldsymbol{z}^{(t)}) \approx \boldsymbol{\epsilon}_k^\top \frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)}) \quad (11)$$

is derived. This is the linear approximation of $L(\boldsymbol{z}^{(t)} + \boldsymbol{\epsilon}_k)$ around $\boldsymbol{z}^{(t)}$. Since the above equation is obtained for all k,

$$\begin{pmatrix} s_1 \\ \vdots \\ s_M \end{pmatrix} \approx \begin{pmatrix} \boldsymbol{\epsilon}_1^\top \\ \vdots \\ \boldsymbol{\epsilon}_M^\top \end{pmatrix} \frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)}) = (\boldsymbol{\epsilon}_1 \ \cdots \ \boldsymbol{\epsilon}_M)^\top \frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)}) \quad (12)$$

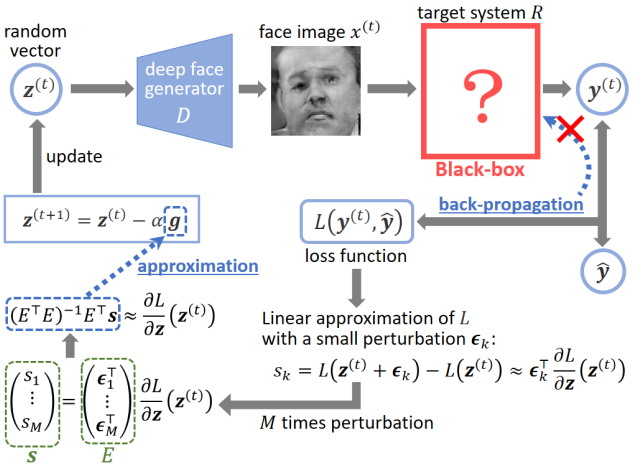

Fig. 3. Overview of proposed MIA method working in black-box setting.

is further derived, where $s_k = L(\boldsymbol{z}^{(t)} + \boldsymbol{\epsilon}_k) - L(\boldsymbol{z}^{(t)})$. Using new symbols $\boldsymbol{s} = (s_1 \ \cdots \ s_M)^\top$ and $E = (\boldsymbol{\epsilon}_1 \ \cdots \ \boldsymbol{\epsilon}_M)^\top$, the above equation is re-written as

$$\boldsymbol{s} \approx E \frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)}) , \quad (13)$$

and therefore $\frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)})$ can be approximated as

$$\boldsymbol{g} = (E^\top E)^{-1} E^\top \boldsymbol{s} \approx \frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)}) . \quad (14)$$

Using the $\boldsymbol{g}$, we update $\boldsymbol{z}^{(t)}$ as

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \alpha \boldsymbol{g} \quad (15)$$

instead of Eq. (7). Fig. 3 shows an overview of the proposed method.

To avoid a singular $E^\top E$, the number of perturbations M should be equal to or larger than the dimension of $\boldsymbol{z}$, namely m. Now we point out another advantage of introducing the deep face generator D. The above approximation method is theoretically applicable without the D, but in this case, we have to make M very large because the dimension of the image domain X is much higher than that of the feature vector domain. Introducing the generator D allows us to use much smaller M, which drastically improves the computational efficiency of the gradient approximation.

C. Criterion for Deciding Perturbation Size

The accuracy of the above approximation method highly depends on the size of the perturbations, namely $||\boldsymbol{\epsilon}_k||$, which is probabilistically decided by σ^2. Larger σ^2 tends to yield larger perturbations, which make the linear approximation of Eq. (11) inaccurate. On the other hand, smaller σ^2 tends to make the perturbations very close to zero, by which the computation of $(E^\top E)^{-1}$ in Eq. (14) numerically unstable. Hence, it is important to appropriately set the value of σ^2.

Fig. 4 shows the relationship between the size of the perturbation and the error of the linear approximation, which

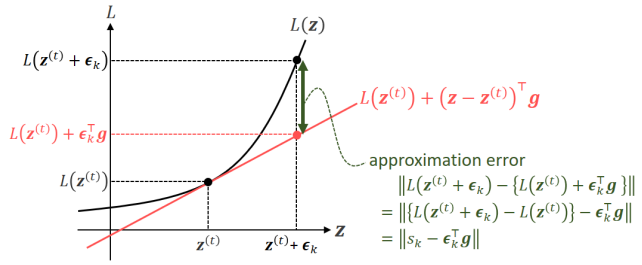

Fig. 4. Relationship between perturbation size and error of linear approximation of Eq. (11)

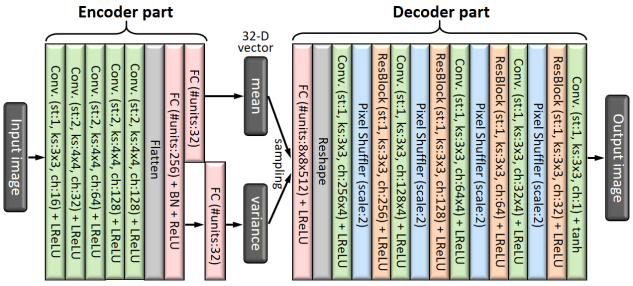

Fig. 5. Network structure of VAE used to train a deep face generator. "Conv." and "ResBlock' 'means convolutional layer and residual block layer, respectively, where "ks", "st", and "ch" mean their kernel size, stride, and num. of channels, respectively. "FC" is fully-connected layer, where "#units" means num. of units. "BN" means batch normalization. "LReLU" is leaky-ReLU activation function.

is calculated as $||s_k - \boldsymbol{\epsilon}_k^\top \boldsymbol{g}||$ when the perturbation size is $||\boldsymbol{\epsilon}_k||$. Since the error should be small for all k, its summation

$$C' = \sum_{k=1}^{M} ||s_k - \boldsymbol{\epsilon}_k^\top \boldsymbol{g}||^2 = ||\boldsymbol{s} - E\boldsymbol{g}||^2 = \boldsymbol{s}^\top (I_M - E(E^\top E)^{-1}E^\top)\boldsymbol{s} \quad (16)$$

seems adequate as a criterion for deciding the perturbation size. However, the above C' becomes very small regardless of the computational stability of the $(E^\top E)^{-1}$ as long as $||\boldsymbol{s}||^2 = \boldsymbol{s}^\top \boldsymbol{s}$ is close to zero. Hence, we divide the C' by $\boldsymbol{s}^\top \boldsymbol{s}$ to regularize it; that is, we propose to use

$$C = \frac{\boldsymbol{s}^\top (I_M - E(E^\top E)^{-1}E^\top)\boldsymbol{s}}{\boldsymbol{s}^\top \boldsymbol{s}} \quad (17)$$

as the actual criterion instead of C'. Specifically, we compute the above C with several different σ^2 and employ the one achieving the smallest C. If C' is fixed, smaller C is obtained from larger $||\boldsymbol{s}||^2$, and the larger $||\boldsymbol{s}||^2$ is provided by larger $||\boldsymbol{\epsilon}_k||$ because of the linear relationship between s_k and $\boldsymbol{\epsilon}_k$. Hence, the above regularization is aiming at larger $||\boldsymbol{\epsilon}_k||$, which ensures the computational stability of the $(E^\top E)^{-1}$.

V. Experiments

This section reports the result of the experiments that we conducted for evaluating the performance of the proposed method.

A. Experimental Setup

1) Dataset: We used the VGGFace2 dataset [22], which consists of around 3.31 million images of 9,131 individuals. Each image contains not only the face region but also the hair and the neck regions captured from various viewpoints. Due to the limitation of our computational resources, we reduced the dataset size by removing non-frontal view images, extracting only the face region from each frontal view image, and performing grayscale transformation on the extracted face regions. As a result, 8,562 indviduals remain, each of whom has at least 5 samples. Then we extracted three subsets from the reduced dataset. These are denoted by Q_1, Q_2, and Q_3 in the remainder of this section. The Q_1 consists of the face images of 4,281 individuals (5 samples per individual), which was used as the attacker's dataset S to train a deep face

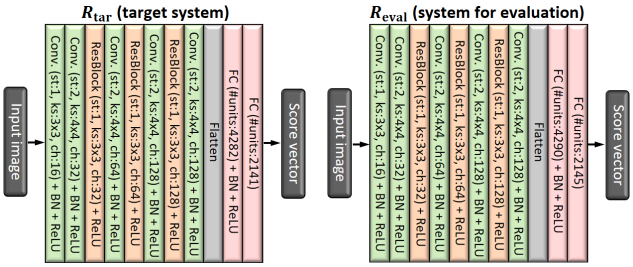

Fig. 6. Network structure of two face recognition systems R_{tar} and R_{eval}.

generator. To make the generator have a good performance, the diversity of face images in S is quite important whereas its size is not so important. Therefore we only used 5 samples per person in the Q_1. On the other hand, the Q_2 and Q_3 consist of the face images of 2,141 and 2,145 individuals (50 samples per individual), respectively, which were used to train two face recognition systems, R_{tar} and R_{eval}. The one trained with the Q_2, i.e., R_{tar}, was the target of MIA. Among its covering 2,141 individuals, we attempted to generate five individuals' face images. We call them "target individuals" in the remainder. The five target individuals are covered also by the Q_3, hence, the face recognition system trained with the Q_3, i.e., R_{eval}, was used for evaluation. In our experiments, we input the resultant images of MIA to the R_{eval} to objectively evaluate the performance of the proposed method. If the resultant images successfully contain the facial characteristics of each target individual, they are expected to be correctly recognized by R_{eval}. Therefore, we employed its recognition accuracy as the objective evaluation criterion.

Note that there is no overlap between the individuals in the Q_1 and those in the Q_2 and Q_3. This is also the case between the Q_2 and Q_3, except for the five target individuals. To reduce the use of computational resources, we converted each face image to grayscale.

2) Network structures: To train the deep face generator D, we employed a variational auto-encoder (VAE) [23], whose decoder part was used as D. Mathematically, the D is a map

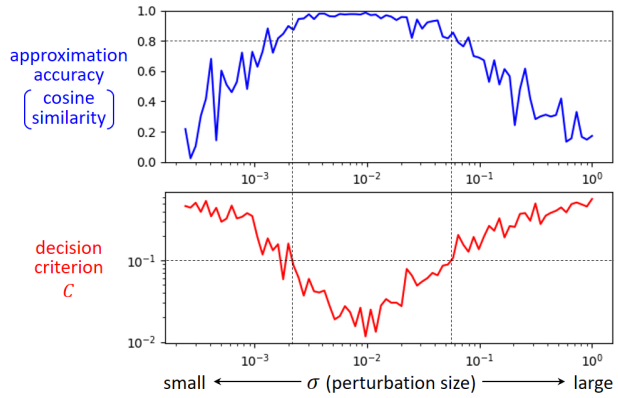

Fig. 7. Relationship between perturbation-size decision criterion C and approximiation accuracy of gradient vectors.

from the latent vector domain to the image domain, whose smoothness provides a positive effect on the performance of the gradient approximation. Therefore we employed VAE here. The network structure of the trained VAE is shown in Fig. 5. We also show the network structure of $R_{\rm tar}$ and $R_{\rm eval}$ in Fig. 6. As seen in this figure, the $R_{\rm tar}$ and $R_{\rm eval}$ have the same structure except for the last two fully-connected layers, but their parameters are different since they were separately trained.

3) Setting of hyper-parameters and seed vector: The VAE trained above was also used for setting a seed vector $\boldsymbol{z}^{(0)}$. For each trial of MIA, we randomly selected 16 images from the attacker's dataset S, namely the subset Q_1, and input them to the encoder part of the VAE to obtain their corresponding feature vectors. Then we calculated their average and used it as $\boldsymbol{z}^{(0)}$.

The proposed method has several hyper-parameters, which were set as below.

- The learning rate α in Eq. (15) : $\alpha = \min\left(100, \frac{0.1}{\|\boldsymbol{g}\|}\right)$.
- The dimension of the feature vector space : $m = 32$.
- The number of the perturbations : $M = 64$.

The σ^2 that is used for deciding the size of the perturbations was chosen from the range of $[10^{-4},\ 10^{0}]$ according to the criterion C in Eq (17). The termination criterion for the iterative process represented by Eq (15) was the same as the one described in Section IV-A.

B. Accuracy of the Gradient Approximation

We first tested the relationship between the criterion C and the accuracy of the gradient vector approximation. To this end, we computed the theoretical value of the gradient $\frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)})$ by Eq (8) as well as its approximation $\boldsymbol{g}$ by Eq (14) in each iteration step t under various settings of σ^2. Then, we measured the cosine similarity between the $\frac{\partial L}{\partial \boldsymbol{z}}(\boldsymbol{z}^{(t)})$ and $\boldsymbol{g}$ to evaluate the approximation accuracy. The higher similarity indicates the better approximation accuracy. Fig. 7 shows the result.

TABLE I
OBJECTIVE EVALUATION: RECOGNITION ACCURACY OF $R_{\rm eval}$. WB-MIA AND BB-MIA MEANS WHITE-BOX MIA AND BLACK-BOX MIA, RESPECTIVELY.

	ID:1	ID:2	ID:3	ID:4	ID:5	Avg.
WB-MIA	86.0%	89.0%	26.0%	79.0%	97.0%	75.4%
BB-MIA	88.0%	82.0%	25.0%	81.0%	99.0%	73.4%

As seen in Fig 7, there is a clear correlation between the approximation accuracy and the criterion C; a higher accuracy can be achieved with a lower C. This result demonstrates that our proposed C is a good criterion for deciding the perturbation size. When $C < 10^{-1} = 0.1$, the approximation accuracy becomes higher than 0.8 in terms of the cosine similarity. We empirically found that this accuracy is enough to make the MIA process successful. The same tendency was found when we used another method such as GAN to train a deep face generator D in an additional experiment, which is not reported in this paper due to space limitations.

C. Performance of the Proposed MIA Method

Next, we examined the performance of the proposed method. In this examination, we also conducted the white-box MIA for the comparison with the proposed method, namely the black-box MIA. For each of the five target individuals mentioned in the previous subsection, we attempted MIA 100 times using different seed vectors. Some examples of the resultant images are shown in Fig. 8 with actual training images of the $R_{\rm tar}$. Not only the images generated by the white-box MIA but also those generated by the black-box MIA look natural and successfully mimic each target individual's facial characteristics contained in the training images. The generated images are a little blurred, which arises from the use of a VAE decoder. A possible solution for the blurring problem is to use a more sophisticated face generator as D.

To quantitatively evaluate the images generated by MIA, we input them to $R_{\rm eval}$ and measured its recognition accuracy, as previously mentioned. The result is shown in Table I, where the recognition accuracy on the images of the black-box MIA is comparable to that of the white-box MIA. This indicates the high performance of the proposed method. Only the face images of the target individual ID:3 are not correctly recognized even in the case of the white-box MIA. We guess the reason is as follows. The subset Q_3, which was used to train $R_{\rm eval}$, contains some other individuals having quite similar facial characteristics to the ID:3 person, whereas the subset Q_2, which was used to train $R_{\rm tar}$, does not contain such individuals. Hence, the images of the ID:3 person tend to be misrecognized only by $R_{\rm eval}$.

In addition to the above objective evaluation, we also conducted a subjective evaluation, where we gave two questionnaires to nine human subjects. In the first questionnaire, we showed 20 images generated by MIA to the nine subjects and asked them who is the person in each image. More specifically, real images of the five target individuals (ID:1-5) and those of two additional dummy individuals were also shown as

Fig. 8. Examples of face images generated by MIA on R_{tar} and its actual training images

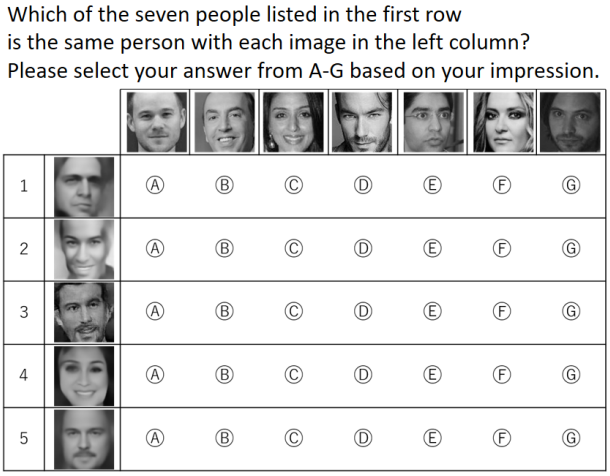

Fig. 9. Example of the questions given to human subjects in the first questionnaire. Images listed in the first row are the real face images of seven individuals A-G. Among them, A, C, D, E, and G are the target individuals of MIA in our experiment. B and F are two dummy individuals, but this fact is not notified to the subjects. On the other hand, images listed in the left column were generated by MIA. For each of them, the subjects try to answer which of A-G is the same person with it.

candidate classes with the 20 generated images. Then, for each of the generated images, the subjects were asked to select the same person with it from the seven candidates, as depicted in Fig 9. The face identification accuracy by the nine subjects is expected to be high if the MIA process succeeded. The result is shown in Table II. In the second questionnaire, we showed 20 pairs of a generated face image and a real face image to the nine subjects and asked them whether the paired images are similar to each other or not. The generated image was made by MIA, aiming at recognizable as the person in its counterpart real image. Hence, the subjects' rate of answering "yes" in this questionnaire is also expected to be high if the MIA process succeeded. The result is shown in Table III.

As shown in Tables II and III, the proposed method of the black-box MIA achieves the performance comparable to the white-box MIA also in the subjective evaluations. The performance for the ID:3 person is not low compared to the other four target individuals, unlike that in the objective evaluation. This fact indicates that many of the images of the ID:3 person generated by MIA are similar to her real face images, but the training set of R_{eval} contains some other individuals having similar facial characteristics, which degrades the recognition performance of R_{eval} for the ID:3 person. On the other hand, the performance for the target individual ID:2 is very low in the subjective evaluations despite his high performance in the objective evaluation. This is because his real face images tend to have a beard, whereas it does not appear in most of the generated images. Essentially, a beard is not a critical characteristic for face identification, but it deeply affects human observers' impressions. Hence, the generated images of the ID:2 person could not be correctly recognized as him by the nine human subjects.

VI. Conclusions

In this paper, we proposed a method for MIA on a black-box face recognition system. MIA can be formulated as a problem of finding the best face image that maximizes the recognition score outputted by a target recognition system. Existing MIA

TABLE II
SUBJECTIVE EVALUATION 1: FACE IDENTIFICATION ACCURACY BY NINE SUBJECTS.

	ID:1	ID:2	ID:3	ID:4	ID:5	Avg.
WB-MIA	94.4%	22.2%	72.2%	33.3%	66.7%	57.8%
BB-MIA	72.2%	16.7%	66.7%	44.4%	66.7%	53.3%

TABLE III
SUBJECTIVE EVALUATION 2: RATE OF ANSWERING THAT MIA-GENERATED IMAGE IS SIMILAR TO REAL ONE.

	ID:1	ID:2	ID:3	ID:4	ID:5	Avg.
WB-MIA	94.4%	11.8%	50.0%	77.8%	72.2%	61.8%
BB-MIA	72.2%	22.2%	70.6%	83.3%	88.9%	67.4%

methods solve this problem by a gradient descent strategy on the assumption that the target system is a white box and its network structure and parameters can be exploited. However, this assumption is not correct in the black-box setting, thus we proposed to numerically approximate the gradient vector in each iteration of the gradient descent by a perturbation-based approach. In the proposed method, it is important for stabilizing the gradient approximation to appropriately decide the size of the perturbations. Hence, we also proposed a decision criterion for it.

We experimentally tested the performance of the proposed method by both the objective and subjective evaluations, whose results demonstrate that the proposed method achieves a comparable performance to the white-box-oriented MIA methods. However, some of the resultant images generated by the proposed method have only insufficient quality; they are partially blurred and lack a beard that is observable in the real face images. We will try to solve these problems in our future work. Another important future issue is to further relax the assumption on a target recognition system. Currently, we assume that the target system outputs the recognition score for all the classes covered by it, although this is not realistic. Therefore, in future work, we will tackle the situation where the attacker can obtain the score only for a few classes.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Numbers JP16H06302 and JP17K00235. This work was also supported by JST, CREST Grant Number JPMJCR20D3, Japan.

REFERENCES

[1] Google Cloud Vision, https://cloud.google.com/vision
[2] Amazon Rekognition Image, https://aws.amazon.com/rekognition/image-features/
[3] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing," in Proc. the 23rd USENIX Security Symposium, pp.17–32, 2014.
[4] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," in Proc. of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp.1322–1333, 2015.
[5] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks," in Proc. the 2020 IEEE Conference on Computer Vision and Pattern Recognition, pp.253–261, 2020.
[6] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar, "Adversarial Machine Learning," in Proc. the 4th ACM Workshop on Security and Artificial Intelligence, pp.43–58, 2011.
[7] M. Kloft and P. Laskov, "Online Anomaly Detection under Adversarial Impact," in Proc. the 13th International Conference on Artificial Intelligence and Statistics, pp.405–412, 2010.
[8] B. Biggio, B. Nelson, and P. Laskov, "Poisoning Attacks against Support Vector Machines," in Proc. the 29th International Conference on Machine Learning, pp.1467–1474, 2012.
[9] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Examples in the Physical World," arXiv preprint, arXiv:1412.6572v3, 14 pages, 2016.
[10] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z.B. Celik, and A. Swami, "Practical Black-Box Attacks against Machine Learning," in Proc. of the 2017 ACM Asia Conference on Computer and Communications Security, pp.506–519, 2017.
[11] F. Tramer, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing Machine Learning Models via Prediction APIs," in Proc. the 25th USENIX Security Symposium, pp.601–618, 2016.
[12] Y. Shi, Y. Sagduyu, and A. Grushin, "How to Steal a Machine Learning Classifier with Deep Learning," in Proc. the 16th IEEE International Symposium on Technologies for Homeland Security, pp.1—5, 2017.
[13] B. Wang and N. Z. Gong, "Stealing Hyperparameters in Machine Learning," in Proc. the 39th IEEE Symposium on Security and Privacy, pp.36–52, 2018.
[14] S. J. Oh, M. Augustin, B. Schiele, and M. Fritz, "Towards Reverse-Engineering Black-Box Neural Networks," Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Springer, pp.121–144, 2019.
[15] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff Nets: Stealing Functionality of Black-Box Models," in Proc. of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, pp.4954–4963, 2019.
[16] M. Juuti, S. Szyller, and S. Marchal, "PRADA: Protecting against DNN Model Stealing Attacks," in Proc. the 2019 IEEE European Symposium on Security and Privacy, pp. 512–527, 2019.
[17] X. Wu, M. Fredrikson, S. Jha, and J. F. Naughton, "A Methodology for Formalizing Model-Inversion Attacks," in Proc. the 29th IEEE Computer Security Foundations Symposium, pp.355–370, 2016.
[18] S. Hidano, T. Murakami, S. Katsumata, S. Kiyomoto, and G. Hanaoka, "Model Inversion Attacks for Prediction Systems: Without Knowledge of Non-Sensitive Attributes," in Proc. the 15th Annual Conference on Privacy, Security and Trust, pp.115–124, 2017.
[19] S. Hidano, T. Murakami, S. Katsumata, S. Kiyomoto, and G. Hanaoka, "Model Inversion Attacks for Online Prediction Systems: Without Knowledge of Non-Sensitive Attributes," IEICE Transactions on Information and Systems, Vol.E101-D, No.11, pp.2665–2676, 2018.
[20] Z. Yang, E. Chang, and Z. Liang, "Adversarial Neural Network Inversion via Auxiliary Knowledge Alignment," arXiv preprint, arXiv:1902.08552, 16 pages, 2019.
[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in Proc. the 27th International Conference on Neural Information Processing Systems, pp.2672–2680, 2014.
[22] Q. Cao, L. Shen, W. Xie, O. Parkhi, and A. Zisserman, "VGGFace2: A Dataset for Recognising Faces across Pose and Age," in Proc. the 13th IEEE International Conference on Automatic Face and Gesture Recognition, pp.67–74, 2018.
[23] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in Proc. the 2nd International Conference on Learning Representations, 14 pages, 2014.