Hierarchical Perceptual Noise Injection for Social Media Fingerprint Privacy Protection

Simin Li, Huangxin Xu, Jiakai Wang, Ruixiao Xu, Aishan Liu, Fazhi He, Member, IEEE, Xianglong Liu, Member, IEEE, and Dacheng Tao, Fellow, IEEE

Abstract— Billions of people share images from their daily lives on social media every day. However, their biometric information (e.g., fingerprints) could be easily stolen from these images. The threat of fingerprint leakage from social media has created a strong desire to anonymize shared images while maintaining image quality, since fingerprints act as a lifelong individual biometric password. To guard the fingerprint leakage, adversarial attack that involves adding imperceptible perturbations to fingerprint images have emerged as a feasible solution. However, existing works of this kind are either weak in black-box transferability or cause the images to have an unnatural appearance. Motivated by the visual perception hierarchy (i.e., high-level perception exploits model-shared semantics that transfer well across models while low-level perception extracts primitive stimuli that result in high visual sensitivity when a suspicious stimulus is provided), we propose FingerSafe, a hierarchical perceptual protective noise injection framework to address the above mentioned problems. For black-box transferability, we inject protective noises into the fingerprint orientation field to perturb the model-shared high-level semantics (i.e., fingerprint ridges). Considering visual naturalness, we suppress the low-level local contrast stimulus by regularizing the response of the Lateral Geniculate Nucleus. Our proposed FingerSafe is the first to provide feasible fingerprint protection in both digital (up to 94.12%) and realistic scenarios (Twitter and Facebook, up to 68.75%). Our code can be found at https://github.com/nlsde-safety-team/FingerSafe.

Index Terms— Fingerprint, adversarial attack, privacy protection.

I. INTRODUCTION

Poster photos on social media is a popular way to share our daily lives with others. However, given that thousands of publicly shared images on Instagram contain accessible fingerprint details [1], personal biometric information (e.g., fingerprints, etc) can be easily stolen from photos shared on social media, which may cause severe security problems for fingerprint authentication systems (e.g., access control system) as shown in Fig.1. There is extensive evidence to support the feasibility of the above challenges. For example, hackers accessed the fingerprint of the president of the EU Commission via an online photo in 2014; recently, it was reported by CNN and BBC that fingerprints have been easily extracted from images shared on social media, even from images taken from 3 meters away. The fingerprint leakage is irreversible—since you cannot change your fingerprint, once the fingerprint has been leaked, all systems that rely on fingerprints are at risk for the rest of your life [2]. Based on these leaked fingerprint images, hackers can gain authorized assess to the access control systems of governments, banks and the police, or payment systems such as ApplePay with more than an 80% success rate with an inkjet-printed paper [3] or a 3D printed mold [4]. The security of fingerprint-based systems is currently at severe risk.

One simple way to protect fingerprint privacy is to erase all fingerprint details from an image (e.g., through masking or pixelization). However, images protected by such methods look very unnatural and are often inappropriate to share on social media (as shown in Fig.1). Moreover, hiding fingerprint details by ad-hoc de-identification methods, such as blurring, provides only weak protection [5]. Recently, adversarial attacks [6], [7] have been used for a benign pro-social purpose, specifically to protect personal privacy by adding small perturbations that are imperceptible to humans but misleading to DNNs [1], [8], [9], [10], [11]. Through the application of adversarial attack strategies to images on social media, hackers

1 https://www.theguardian.com/technology/2014/dec/30/hacker-fakes-german-ministers-fingerprints-using-photos-of-her-hands
2 https://www.bbc.com/news/uk-wales-4371477
3 https://edition.cnn.com/2021/05/25/uk/drug-dealer-cheese-sentenced-scl-gbr-intl/index.html
4 https://www.ft.com/content/446ac9a2-9e21-11ed-9d7c-be108f1c1dce
5 https://www.forbes.com/sites/zakoffman/2019/08/14/new-data-breach-has-exposed-millions-of-fingerprint-and-facial-recognition-records-report/?sh=38315c146c60
6 https://arstechnica.com/information-technology/2020/04/attackers-can-bypass-fingerprint-authentication-with-an-80-success-rate/
Moreover, with regard to visual naturalness, local contrast (a low-level stimulus) triggers the largest unnaturalness response in humans [14]. We therefore suppress the low-level local contrast using the response of the ganglion cells of the retina and the Lateral Geniculate Nucleus (LGN) in the human visual system [16] to promote visual naturalness. The contributions of this paper can be summarized as follows:

- To the best of our knowledge, FingerSafe is the first work to achieve strong black-box protection capability (e.g., unknown architectures, preprocessors, etc.) in fingerprint privacy protection. In contrast, all baseline methods failed when presented with unknown preprocessing.
- We propose an orientation distortion module (high-level semantics) and a local contrast suppression module (low-level stimulus), enabling FingerSafe to achieve better transferability and naturalness results simultaneously.
- Extensive experiments demonstrate that FingerSafe outperforms baselines by large margins in digital and real-world scenarios (Twitter and Facebook).

II. RELATED WORKS

In this section, we briefly review related works in fingerprint recognition, adversarial attacks, and privacy protection.

A. Fingerprint Recognition

A fingerprint recognition system primarily consists of four basic steps: capturing biometric fingerprint data, preprocessing, feature extraction and comparison. Preprocessing algorithms are designed to retain the ridge features of fingerprint and filter out noisy and spurious features [17], [18], [19]. Afterwards, discriminative fingerprint features were extracted from preprocessed images. Traditional feature extraction methods are usually either minutiae-based (e.g., ridge ending, bifurcation, and short ridge) [20], [21] or local descriptor-based [22], [23], [24]. Recently, the use of neural networks for feature extraction has has been introduced [25], [26], [27]. Finally, comparison characterizes the purpose of recognition: for fingerprint verification, the goal is to determine if the two fingerprints are from the same finger [17], [28]; for fingerprint identification, the goal is to search for possible defenses [36], [37], [38], [39], [40], [41]. In general, adversarial attack methods can be categorized into white-box attacks [1], [9], [42] (attackers have direct access to the structure and parameters of the model) and black-box attacks [10], [13], [43] (attackers have limited knowledge...
of the model; for example, the architecture or operations may be unknown). In this paper, we primarily focus on black-box attacks, i.e., using adversarial attacks to protect fingerprint privacy in the black-box setting.

C. Privacy Protection

Privacy protection aims at making it impossible for hackers to exploit personal data from acquired content. A simple solution for privacy protection is to obfuscate the image (e.g., blurring, pixelation, darkening [44]). However, these methods are either ineffective [45, 46] or visually unsatisfying [1], making them unsuitable for use in protecting fingerprint privacy on social media. Recently, several works have been developed that aim to protect biometric privacy using noise that is imperceptible to humans. Unlearnable Examples [8] was proposed to protect privacy by fooling models into learning meaningless features during training. To protect facial images, Fawkes [9] was designed to generate noise that creates maximal changes in image representation. Similarly, LowKey [11] used an ensemble strategy and added a loss term to Fawkes in order to additionally perturb image representation under Gaussian smoothing. TIP-IM [10] protected facial images by generating adversarial identity masks comprised of many target identities. For fingerprint protection, [1] is the only research in the white-box setting. However, their white-box protection performance is limited: hackers can sidestep this white-box protection by simply using another fingerprint recognition method.

In contrast to previous studies, our FingerSafe achieves strong black-box transferability and naturalness, meaning that it can be used to protect fingerprint privacy in complex real-world scenarios (e.g., those in which hackers have different goals and use different learning algorithms, model architectures, etc).

III. APPROACH

A. Problem Definition

Given an image x containing a fingerprint, a DNN model \( F \) is designed for fingerprint recognition on the preprocessed fingerprint \( P(x) \) such that \( F(P(x)) = y \), here, \( P \) is the image preprocessing algorithm and \( y \) is the ground-truth label of \( x \) (i.e., person ID).

In real-world scenarios, hackers could leverage the leaked image \( x \) to sneak past the third-party fingerprint recognition system \( F_\theta \), where we call it testing stage attack. Correspondingly, in testing stage protection, we aim to protect the abuse of fingerprint privacy by the hackers. More specifically, we generate adversarial examples \( x_{adv} \) that are visually similar to clean examples \( x \), but prevent hackers from gaining unauthorized access in \( F_\theta \):

\[
F_\theta(P(x_{adv})) \neq y \quad s.t. \quad \|x - x_{adv}\| \leq \epsilon, \quad (1)
\]

where \( \|\cdot\| \) is a distance metric (e.g., \( \ell_1, \ell_2, \ell_\infty \)) to constrain the difference between \( x \) and \( x_{adv} \). In this paper, we refer to \( x_{adv} \) as protected images or adversarial examples interchangeably.

We also highlight our fingerprint privacy protection problem with biometric template protection (BTP) [47, 48, 49, 50], [51]. BTP seeks to extract a secure template (i.e., representation) from sensitive biometric data, such that hackers cannot extract personal identifiable information or reverse-engineer the original fingerprint image. The safe representation can then be used for fingerprint recognition. Our fingerprint privacy protection differs from BTP in two aspects. First, our FingerSafe adds small and natural noise to the fingerprint image, instead of representation. Second, unlike safe representation that can be used for fingerprint recognition, images protected by FingerSafe cannot be used by a hacker to pass different types of fingerprint recognition systems.

In practice, protectors are not aware of which model is being used by the third-party authority (i.e., \( F_\theta \) is a black-box model), while different authorities might also use different models. We thus generate the protected image \( x_{adv} \) from a surrogate model \( F_\theta \) and transfer it to the unknown target model \( F_\theta \).

In this paper, we focus on testing stage protection. However, we find that FingerSafe is also effective in training stage protection, where the goal of hackers is to collect \( x \) and use it to train a highly accurate model \( F_\theta \) for commercial use. Note that \( F_\theta \) is the model trained by hackers, while \( F_\theta \) is the model used by an authorized third party.

B. FingerSafe Framework

Our work is built around the concept of the visual perception hierarchy, which holds that different levels of perception focus on different abstract stages and make different contributions to model perception and human perception. As shown in Fig. 2, our FingerSafe framework consists of two main modules, namely an Orientation field distortion module and a Local contrast suppression module, which respectively enforce strong protection transferability by attacking high-level semantics and enforce high naturalness by preserving low-level stimulus.

Regarding protection transferability, high-level perception focuses on predictions using abstract semantics that is shared between models. Similarly, perturbing these semantics is a form of model-agnostic attack that transfers well between different models. Specifically, we propose to estimate and distort the fingerprint orientation field, a high-level feature of the fingerprint. When this orientation field is distorted, the protected fingerprint is semantically different from the previous raw image, making this an approach that can be naturally transferred across different goals, model architectures, and preprocessing methods.

Regarding visual naturalness, motivated by the stimulus-response theory in psychology, humans judge whether something appears unnatural by assessing the local contrast, which is a low-level stimulus. Clearly, naturalness will be enhanced if suspicious low-level stimuli are not present. We first calculate how the stimuli will appear to the human visual system, then add a regularization term to suppress the sensitive stimuli (local contrast) in FingerSafe, creating protected images that are also suitable for sharing on social media.

C. Orientation Field Distortion

Previous works have found that high-level features reflect the discriminative semantics towards a specific class, which are
shared between models [12], [13], [52], [53]. Thus, attacking the model-shared high-level semantics could improve the transferability of the attack among models. In the context of fingerprint recognition, a ridge feature is a high-level representation that reflects the uniqueness of a fingerprint, can be used to determine identity [54], and is widely considered to be invariant over a person’s lifetime. Therefore, ridge features are often shared by different models, and recognition approaches are built upon these preprocessed ridge features obtained from clean fingerprints [15], [18], [19].

However, directly attacking ridge features is difficult because the ridge features \( P(x) \) are hidden in fingerprint image \( x \). Moreover, the model (including preprocessing algorithms) used to extract the ridge features is unknown to us. To reliably perturb ridge features, we inject protective noise into the fingerprint orientation field, an estimate of the pixelwise orientation of ridge feature \( P(x) \) that reflects the intrinsic traits of fingerprints and is agnostic to different models. Due to their close relationship, distorting the orientation field will force a change in the underlying ridge features. As a result, different models that rely on ridge features will fail to make correct predictions when false ridge features are provided.

Specifically, we maximize the differences in the orientation field between clean image \( x \) and protected image \( x_{adv} \) using least mean square orientation estimation [15] with an orientation estimation module \( \Phi \) as follows:

\[
\phi = \Phi(x) = \frac{1}{2} \tan^{-1} \frac{G \ast (2G_x \ast x \odot G_y \ast x)}{G \ast ((G_x \ast x)^2 - (G_y \ast x)^2)} + \frac{\pi}{2},
\]

where \( \ast \) is the convolution operator, \( \odot \) is the pointwise Hadamard product, \( G_x \) and \( G_y \) are Gaussian derivative kernels at direction \( x \) and \( y \), and \( G \) is a Gaussian kernel used to average the orientation estimates and avoid ambiguity.

Given the orientation field \( \phi_{adv} \) calculated by Eqn. 2, we introduce the following orientation field distortion loss, \( L_O \):

\[
L_O = -\frac{1}{HW} \sum_h \sum_w \| \sin(\phi_{adv}^{h,w} - \phi^{h,w}) \|_1,
\]

where \( \| \cdot \|_1 \) is the \( \ell_1 \) norm, while \( H \) and \( W \) are the height and width of image \( x \). Since \( \phi_{adv} \) and \( \phi \) are angles, we use a \( \sin(\cdot) \) function to measure their difference.

Additionally, we adopt an adversarial loss \( L_{adv} \) to further guide the protective noise generation process. \( L_{adv} \) maximizes the distance of the representation calculated from our source model \( F_\theta \) between the protected image and clean image as follows:

\[
L_{adv} = -\| F_\theta(x_{adv}) - F_\theta(x) \|_2.\]

### D. Local Contrast Suppression

To achieve reliable privacy protection, existing works usually generate attacks with large perturbation budgets [10], [11], which result in a suspicious appearance and reduced utility. Motivated by extensive biological observations suggesting that human vision is highly sensitive to the subtle variation of low-level stimuli (i.e., local contrast), we therefore generate protected images with higher naturalness by preserving these sensitive features intact.

Specifically, local contrast perceived by human is characterized by the ganglion cells of the retina and the Lateral Geniculate Nucleus (LGN) neurons. Their responses could be calculated by using a modified differences of Gaussian (DOG) model [16]. We therefore introduce a local contrast calculation module \( \Omega \) as follows:

\[
G_c(i, j) = \exp \left[ -(i/r_c)^2 - (j/r_c)^2 \right],
\]

\[
G_s(i, j) = 0.85 (r_c/r_s)^2 \exp \left[ -(i/r_s)^2 - (j/r_s)^2 \right],
\]

\[
\omega = \Omega(x) = \frac{G_c \ast x - G_s \ast x}{G_c \ast x + G_s \ast x},
\]

where \( G_c \) and \( G_s \) are two Gaussian kernels that calculate the center and surrounding component with receptive fields \( r_c \) and \( r_s \). \( \Omega(x) \) is simply a DOG model with the term \( G_c \ast x + G_s \ast x \) to simulate the light adaptation process of ganglion cells of the retina and LGN.

However, simply keeping local contrast intact will have the opposite result to fingerprint orientation field distortion, and may thus fail to provide protection. Thus, to release the constraint of regularizing local contrast on all parts of the image, we propose a local contrast attention module \( \Phi \) to identify the importance of different regions within an image [55] and thereby inject different perturbation budgets,
which can be expressed as follows:
\[
\psi = \Psi(x) = G \ast F^{-1} \left[ \exp(I - B \ast I) + \mathcal{P} \right]^2, \tag{6}
\]
where \( A = Re(F(x)) \), \( \mathcal{P} = Im(F(x)) \) and \( I = \log(A) \) are the amplitude spectrum, phase spectrum, and logarithm of the amplitude spectrum. \( F \) and \( F^{-1} \) denote the Fourier and inverse Fourier transform, \( G \) and \( B \) are a Gaussian kernel and box kernel, respectively. Note that \( \psi_{\text{adv}} \) was used to identify important areas in both clean and protected images, while \( H \) and \( W \) are the height and width of image \( x \). Thus, the local contrast suppression loss \( L_C \) can be written as follows:

\[
L_C = \frac{1}{HW} \sum_{h} \sum_{w} ReLU \left( \left( \alpha_{h,w}^{\text{adv}} - \alpha_{h,w}^L \right) \otimes \psi_{h,w}^L \right). \tag{7}
\]

E. Overall Training

Overall, we generate the protected image by jointly optimizing the adversarial loss \( L_{\text{adv}} \), orientation field distortion loss \( L_O \), and local contrast suppression loss \( L_C \). The implementation of \( L_O \) and \( L_C \) includes only Gaussian smoothing and Fast Fourier Transform. As a result, the execution time of FingerSafe is similar to that of the state-of-the-art adversarial attacks [42]. Using an iterative gradient-based method [42], we can generate the transferable and visually natural protected image \( x_{\text{adv}} \) by minimizing the formulation below:

\[
\min_{x_{\text{adv}}} L_{\text{adv}} + \lambda L_O + \gamma L_C, \tag{8}
\]

where \( \lambda \) and \( \gamma \) are hyperparameters to control the strength of \( L_O \) and \( L_C \), respectively. \( clip(x, a, b) \) is the clip function that clips the value of \( x \) to the minimum value of \( a \) and maximum value of \( b \), as we assume \( x_{\text{adv}} \) is bounded by \( \ell_{\infty} \) norm, following the standard setting in PGD attack [42]. To balance the attacking ability \( L_{\text{adv}} \), transferability \( L_O \), and visual naturalness \( L_C \), we set \( \lambda \) as \( 10^2 \) and \( \gamma \) as \( 5 \times 10^2 \); notably, a small \( L_O \) cannot sufficiently perturb the high-level semantics, while a high \( L_C \) will erase the protection pattern entirely. Note that these parameters, \( \lambda \) and \( \gamma \), work well for the two datasets used in our experiments. For details, please see our ablation studies in Section IV-F. The overall training algorithm for FingerSafe is presented in Algorithm 1.

The number of iterations \( N \) is the number of iterations in the iterative gradient-based method, set to 20 throughout this paper. \( N \) is rooted in the principles of PGD [42]. Technically, \( N \) signifies the maximum number of steps permissible within our FingerSafe algorithm. In essence, it denotes for each input image, the number of iterations we compute gradients from the DNN and apply them to the input image to produce a protected image for each iteration. Once \( N \) iterations are completed, the algorithm stops for the current input image, storing the protected image, and proceeds to the subsequent image. The step size is set to \( \alpha \) as \( \epsilon / 10 \), in accordance with the default setting of PGD [42]. For experiments regarding the selection of the step number \( N \) and step size \( \alpha \), refer to Section IV-E. 2) and 3).

To further elaborate, let’s revisit the initial adversarial attack, FGSM [7], which operates through a onestep gradient ascent process: \( x_{\text{adv}} \leftarrow x + \epsilon \text{sgn} (\nabla_{x}(L_{\text{adv}} + \lambda L_O + \gamma L_C)) \) with perturbation budget \( \epsilon \) bounded by \( \ell_{\infty} \) norm. This essentially linearizes the loss function \( L_{\text{adv}} + \lambda L_O + \gamma L_C \) with respect to the current network parameters \( \theta \). However, this one-step approach can be imprecise and has subsequently been replaced by a multi-step, more potent variant widely recognized as the PGD attack [42], an adversarial attack using projected gradient descent to generate the adversarial noise. In mathematical terms, it is represented as \( x_{\text{adv}} \leftarrow \prod_{i=0}^{N-1} \text{clip}(x_{\text{adv}} + \epsilon \text{sgn}(\nabla_{x}(L_{\text{adv}} + \lambda L_O + \gamma L_C)), x-\epsilon, x+\epsilon) \), where \( x_{\text{adv}} = x \). The PGD attack employs an iterative projected gradient descent technique, utilizing a smaller step size \( \alpha \) and a greater number of iterations.

\begin{algorithm}[h]
\caption{FingerSafe Privacy Protection}
\begin{algorithmic}[1]
\State \textbf{Input:} Fingerprint image database \( \mathcal{D} \), orientation estimation module \( \Phi \), local contrast estimation module \( \hat{\Omega} \), saliency map estimation module \( \Psi \).
\State \textbf{Output:} Protected image database \( \mathcal{D}_{\text{adv}} \)
\For {Minibatch \( x \) in dataset \( \mathcal{D} \)}
\State \( x_{\text{adv}}^0 = x \), \( i = 0 \)
\For {in number of iterations \( N \)}
\State \( \phi \leftarrow \Phi(x) \), \( \phi_{\text{adv}}^i \leftarrow \Phi(x_{\text{adv}}^i) \) by Eqn. 2
\State \( \omega \leftarrow \Omega(x) \), \( \omega_{\text{adv}} \leftarrow \Omega(x_{\text{adv}}) \) by Eqn. 5
\State \( \psi \leftarrow \Psi(x) \), \( \psi_{\text{adv}} \leftarrow \Psi(x_{\text{adv}}) \) by Eqn. 6.
\State calculate \( L_O, L_C \) and \( L_{\text{adv}} \) by Eqn. 3, 7, 4.
\State \( x_{\text{adv}}^{i+1} \leftarrow \text{clip}(x_{\text{adv}}^i + \alpha \text{sgn}(\nabla_{x}(L_{\text{adv}} + \lambda L_O + \gamma L_C)), x-\epsilon, x+\epsilon) \)
\EndFor
\State \( \mathcal{D}_{\text{adv}} \leftarrow x_{\text{adv}}^{i+1} \)
\EndFor
\end{algorithmic}
\end{algorithm}

Theoretical Aspects: Our algorithm can be viewed as a specific instance of the general-purpose PGD algorithm, designed for solving constrained optimization problems. In response to reviewer suggestions, we provide a concise overview of the theoretical properties of PGD and also our algorithm.

Our primary aim is to minimize the following objective function: \( \min_{\|x_{\text{adv}} - x\|_{\infty} \leq \epsilon} (L_{\text{adv}} + \lambda L_O + \gamma L_C) \). This objective can be regarded as a constrained optimization problem: \( \min_{x \in \Omega} F(x) \). Projected Gradient Descent (PGD) has been proposed as an effective method for addressing such problems [56]. It is represented as follows: \( x_{i+1} = P(x_i - \alpha \nabla_x F(x_i)) \), where \( P \) represents the projection of \( x \) onto the constrained space \( \Omega \). Interestingly, this very same algorithm is widely employed in adversarial attacks under the name PGD attack [42]. In a similar vein, our algorithm, \( x_{i+1} \leftarrow \text{clip}(x_{\text{adv}} + \epsilon \text{sgn}(\nabla_{x}(L_{\text{adv}} + \lambda L_O + \gamma L_C)), x-\epsilon, x+\epsilon) \), is an instantiation of PGD. Here, the projection function \( P \) is the clip function, and the objective function \( F(x) \) is \( -L_{\text{adv}} + \lambda L_O + \gamma L_C \).

The convergence of our algorithm aligns directly with the principles of PGD. As indicated in [56], when \( F(x) \) is convex, bounded, and differentiable, and when \( \nabla_x F(x) \) exhibits Lipschitz continuity with a Lipschitz constant \( L \) (meaning, for all valid \( x \), there exists a constant \( L \) that satisfies \( ||\nabla_x F(x) - \nabla_x F(y)|| \leq L ||x - y|| \)).
\[ \nabla_y F(y) \leq L||x - y||, \] further when the (fixed) step size adheres to \(0 < \alpha < 2/L\), the PGD algorithm assures that the subsequence of \(x_i\) will weakly converge to the optimum \(x^*\) at a rate of \(O(i^{-1})\), denoted as \(F(x_i) - F(x^*) \leq c/i\), with \(c\) representing a constant.

This theoretical framework yields two essential insights. Firstly, while the step size \(\alpha\) can be determined within the range \(0 < \alpha < 2/L\), estimating the Lipschitz constant for Deep Neural Networks (DNNs) proves to be challenging. The requirement for global Lipschitz continuity implies the necessity to evaluate \(F(x)\) for all possible inputs, a task known to be NP-hard [57]. Consequently, we must rely on empirical approximations. The key lesson here is that \(\alpha\) should be sufficiently small to ensure convergence but not excessively so, as an overly small \(\alpha\) would prolong the convergence process unnecessarily. Following the established settings in [42], we have empirically found that \(\alpha = \epsilon/10\) performs effectively in practice. Secondly, with a convergence rate of \(O(i^{-1})\), the speed of convergence is inversely proportional to the number of iterations, denoted as \(i\). This implies that as we increase the number of iterations, the optimization algorithm’s progress towards the optima becomes progressively slower. This helps rationalize the selection of the number of iterations: due to the \(O(i^{-1})\) convergence rate, the addition of more iterations \(i\) contributes less and less to convergence as \(i\) increases. Consequently, we have empirically set the number of iterations to 20 since this allows all methods to achieve full convergence in practical applications.

IV. EXPERIMENTS

In this section, we first describe our experimental settings, then evaluate the transferability, naturalness, and real-world social media protection ability of FingerSafe.

A. Experimental Setup

1) Dataset and Target Models: We conduct our experiment on the HKPolyU Database [58], a widely used fingerprint dataset containing photographs of fingerprints from 336 different subjects. While most images in social media are in color, the given dataset is in grayscale. We use an off-the-shelf image colorization method [59] to transform the whole dataset into RGB format.

Regarding fingerprint recognition, for the deep learning-based methods, we follow the settings presented in [1] and [60], which first extracts fingerprint features using a DNN model and then pairs the images using Euclidean distance. We consider two common tasks in fingerprint recognition, including fingerprint verification and identification. Despite both methods rely on comparison of extracted features, they focus on different tasks and are treated separately in literature [17], [61], [62]. Fingerprint verification is a one-to-one process that determines if a pair of fingerprints belongs to the same person, the goal is to determine the task with high confidence. In contrast, fingerprint identification is a one-to-many process which finds the identity with given fingerprint, the goal is to provide a fast retrieval of the input fingerprint in a large database.

For DNNs, we use ResNet50 [63], Inceptionv3 [64] and DenseNet121 [65] as feature extractors. For traditional feature extraction methods, we use ScatNet [24] and FingerNet [25], where traditional discriminant features are extracted from fingerprint images, then used for subsequent recognition. The preprocessing methods we use include Sanharan et al. [18], Lin et al. [18], and Frangi filter [66]. In the remainder of this paper, we use MHS (Median filtering, Histogram equalization and Sharpening) as shorthand for the method of Sanharan et al. and HG (Histogram equalization and Gabor filtering) to denote the method of Sanharan et al.

2) Compared Methods and Evaluation Metrics: For attack methods that protect privacy, we choose the current state-of-the-art methods, namely PGD [42], Unlearnable Example (denoted Unlearn.) [8], Fawkes [9], LowKey [11] and Malhotra et al. [1] (denoted as Malhotra). Note that we select PGD as a representative of state-of-the-art general-purposed adversarial attack, thus do not compare with early methods such as FGSM [7]. Empirically, we also find FGSM yields inferior result. See supplementary materials for more details. Among these baselines, Unlearnable examples are designed for training stage protection, while others aim at testing stage protection. For fair comparison, FingerSafe and all baselines use the same set of hyperparameters if possible. We use their released code for all baselines. For methods with own hyperparameters (e.g., LowKey), we tune their results for best performance. See more details in supplementary materials.

In particular, for testing stage protection, we assume that hackers collect fingerprints from the internet and attempt to sneak past a well-trained authentic fingerprint recognition system with a clean fingerprint database. As a result, we train the recognition model using clean image pairs, then use the clean-protected image pairs to evaluate the model accuracy. These systems typically select a fixed threshold for fingerprint comparison that determines whether two fingerprints match. Since authentic recognition systems have no reason to turn against our protection, we assume the threshold of the system is fixed in our main experiment, while we also discuss the scenario of using different thresholds in transferability of FingerSafe section. For training stage protection, we assume that hackers collect additional fingerprints from website to enhance the performance of their fingerprint recognition system, like clearview.ai [67]. Thus, we train the recognition model using protected image pairs, then use clean image pairs to verify the model performance.

Consequently, to evaluate the result of FingerSafe, we use model accuracy (ACC) as the evaluation metric for identification. For verification, we select the threshold of fingerprint comparison as the best clean threshold and use True Positive Rate (TPR) as evaluation metric, which measures the probability of hackers successfully sneaking in fingerprint recognition system. For verification with non-fixed thresholds, we evaluate the performance using Equal Error Rate (EER) and Detection Error Tradeoff (DET) plot.

3) Implementation Details: In the orientation estimation module \(\Phi, G_\phi\) and \(G_{\phi, \gamma}\) are \(7 \times 7\) derivatives of the Gaussian kernel in the \(x\) and \(y\) directions. The Gaussian kernel \(G\) has size \(31 \times 31\). In the local contrast calculation module \(\Omega, r_\epsilon\) and \(r_s\)
are set to 2 and 4, while the sizes of $G_G$ and $G_B$ are 13*13 and 25*25, respectively. In the saliency map calculation module $\Psi$, $G$ is a Gaussian kernel with size 9*9 and $B$ is a box kernel with size 3*3. These kernel sizes and parameters follow the original implementation. For the compared methods, we set the dissimilarity threshold to 0.1 to generate Fawkes adversarial samples, and further set the parameter of LPIPS to 5 to generate Fawkes adversarial samples. When extracting the ScatNet feature, the number of scales $J$ and orientations $L$ are set to 2 and 8, respectively, with input shape of (50, 50).

We empirically set $\gamma = 10^2$, $\lambda = 5 \times 10^2$ for protected image generation. Empirically, we have found that this set of parameters performs well across the two datasets used in our experiments. For detailed information, please refer to our ablation studies in Section IV-F. The number of iterations $N$ is set to 20 for FingerSafe and all iterative gradient based attacks, the step size is set to $\alpha = \epsilon / 10$ following the default setting of PGD [42]. Since Fawkes is an optimization based method [34], the number of iterations is set to 150, following their strongest default setting. All fingerprint recognition models are optimized by an Adam optimizer with a learning rate of $10^{-4}$ and a maximum of 50 epochs. We keep the magnitude of perturbations the same for all methods at $\epsilon = 8/255$ in terms of $\ell_\infty$-norm. All of our codes are implemented in PyTorch [68]. We conduct all experiments on a NVIDIA Tesla V100-SXM2-16GB GPU cluster.

For all of our experiments, the fingerprint images are either obtained from the public HKPolyU dataset under a license agreement or collected from participants who signed an agreement with us under supervision of an ethic committee that their information will only be used for non-commercial research.

### B. Transferability of FingerSafe

In this section, we conduct extensive experiments to evaluate the transferability of FingerSafe in black-box settings using multiple model architectures, preprocessors, and traditional feature extraction methods. We show the performance of different purposes (verification and identification, denoted as Veri. and Iden.) in each case.

#### 1) Different Deep Learning Architectures: We start from the case in which hackers use black-box DNN models for recognition, and FingerSafe aims to generate images that are capable of transferring between different black-box architectures. For fair comparison, we keep other settings the same for different architectures. Here, we use ResNet50, InceptionV3, DenseNet121, and an ensemble of these models (used in Lowkey) as our source model to generate adversarial examples, and further use ResNet50, InceptionV3, and DenseNet121 as the target models (the case in which the source and target models are the same represents a white-box attack, while that in which these models are different represents a black-box attack). From the results in Tab.I, we can draw several conclusions, as follows:

1. For black-box attacks, FingerSafe consistently demonstrates superior transferability and outperforms LowKey, the best-performing baseline, by large margins (i.e., up to 7.71% protection improvement over the best-performing method).
2. For white-box attacks ($\text{ResNet50} \rightarrow \text{ResNet50}$, $\text{InceptionV3} \rightarrow \text{InceptionV3}$, $\text{DenseNet121} \rightarrow \text{DenseNet121}$), FingerSafe also consistently provides stronger protection, outperforming Unlearn. by large margins in $\text{InceptionV3} \rightarrow \text{InceptionV3}$ (i.e., up to 7.84% protection improvement over the best-performing method).
3. Compared with FingerSafe, the baselines do not transfer well to different models, especially InceptionV3. This can be explained with reference to the noise in Fig. 3: the noise generated by the baselines does not contain a semantic-related component, while the protective noise of FingerSafe contains fingerprint-alike high-level semantics, which are shared by different DNN architectures. Thus, protected images generated by FingerSafe are likely to be viewed as semantically different fingerprints, which is model-agnostic.

| From/To | Task | Veri. TPR (%) | Iden. ACC (%) |
|---------|------|---------------|---------------|
|         | Method | ResNet50 | InceptionV3 | DenseNet121 | ResNet50 | InceptionV3 | DenseNet121 |
|         | PGD | 3.48 | 43.28 | 3.36 | 10.29 | 33.82 | 12.50 |
|         | Unlearn. | 5.58 | 54.73 | 5.85 | 16.18 | 41.18 | 18.38 |
|         | Fawkes | 0.00 | 15.30 | 2.11 | 0.00 | 10.29 | 7.35 |
|         | Malhotta | 5.35 | 50.50 | 13.31 | 12.50 | 46.32 | 35.29 |
|         | FingerSafe | 0.00 | 3.86 | 1.62 | 0.00 | 5.88 | 2.21 |
| ResNet50 | PGD | 24.75 | 12.19 | 23.76 | 28.67 | 20.59 | 30.15 |
|         | Unlearn. | 1.74 | 11.07 | 2.99 | 8.09 | 5.15 | 7.35 |
|         | Fawkes | 0.50 | 15.30 | 4.60 | 6.62 | 3.23 | 6.22 |
|         | Malhotta | 1.99 | 14.47 | 22.61 | 7.35 | 24.26 | 29.41 |
|         | FingerSafe | 0.12 | 3.23 | 0.37 | 3.68 | 2.21 | 2.21 |
| InceptionV3 | PGD | 10.32 | 57.96 | 0.25 | 15.44 | 44.85 | 5.88 |
|         | Unlearn. | 2.61 | 31.09 | 0.12 | 5.15 | 4.41 | 2.94 |
|         | Fawkes | 0.37 | 14.55 | 0.00 | 6.62 | 9.56 | 1.47 |
|         | Malhotta | 2.49 | 19.03 | 2.11 | 2.94 | 5.88 | 2.21 |
|         | FingerSafe | 0.00 | 4.73 | 0.00 | 2.94 | 2.21 | 0.00 |
| DenseNet121 | PGD | 10.32 | 57.96 | 0.25 | 15.44 | 44.85 | 5.88 |
|         | Unlearn. | 2.61 | 31.09 | 0.12 | 5.15 | 4.41 | 2.94 |
|         | Fawkes | 0.37 | 14.55 | 0.00 | 6.62 | 9.56 | 1.47 |
|         | Malhotta | 2.49 | 19.03 | 2.11 | 2.94 | 5.88 | 2.21 |
|         | FingerSafe | 0.00 | 4.73 | 0.00 | 2.94 | 2.21 | 0.00 |
| Ensemble | LowKey | 0.87 | 11.57 | 3.23 | 3.68 | 7.35 | 3.70 |
Fig. 3. Visualization of protected images and protective noises under the same protection capability. The noise of FingerSafe contains fingerprint-like high-level semantic and achieves stronger transferability under various transfer settings.

| Preprocessor | Clean | PGD | Unlearn. | Fawkes | Lowkey | Malhotra | FingerSafe |
|--------------|-------|-----|---------|--------|--------|----------|------------|
| MHS          | 93.03 | 68.28 | 67.54 | 69.43 | 77.74 | 59.33 | 5.85      |
| HG           | 88.06 | 82.90 | 82.09 | 73.51 | 80.72 | 81.09 | 4.60      |
| Frangi       | 89.43 | 22.01 | 24.88 | 7.34  | 51.99 | 32.09 | 6.59      |

2) Different Preprocessors: We further assume that hackers employ unknown preprocessors to extract ridge features and use them for further recognition, which is common in fingerprint processing [17], [18], [19], [66]. To further validate the effectiveness of our FingerSafe in this more practical scenario, we generate protected images on a ResNet50 source model without preprocessors, and then transfer to a target ResNet50 model that uses a variety of preprocessors (including MHS, HG, and Frangi filter). Note that FingerSafe achieves similar strong protection on different source models. Here we use ResNet50 since it is the most commonly used backbone in computer vision, see more details in Discussion section. As shown in Tab. II, FingerSafe achieves significantly strong performance and outperforms others by large margins when preprocessors are used for target models (i.e., up to 68.91% protection improvement over the best-performing method). Importantly, the baseline methods are ineffective in this scenario.

To better understand the performance of FingerSafe in this setting, we visualize the fingerprints extracted by all preprocessing methods in Fig. 4. As discussed in section III-C, the goal of preprocessing is to extract high-level ridge features with rich semantics, while discarding low-level features. As a consequence, the noise generated by baselines contains both high-level ridge features and low-level non-ridge features. After preprocessing, all non-ridge noises are simply filtered out (i.e., baselines failed to attack the high-level semantics). In contrast, the protective noise of FingerSafe contains high-level, ridge-like patterns that can be used to attack different preprocessors. Thus, hackers cannot even extract the true fingerprint from images protected by FingerSafe, let alone use it for correct recognition.

3) Traditional Feature Extraction Methods: In this section, we assume that hackers use traditional feature extraction methods, such that fingerprint comparison is performed based on traditional features (i.e., minutiae, ScatNet features, etc.). In more detail, we generate protected images on a ResNet50 source model and transfer it to Scattering Network (ScatNet) [23] and a minutiae-based comparison [25]. Specifically, we use $\ell_2$ distance as the similarity between two ScatNet features. For minutiae comparison, we extract the minutiae by [25] and employ the method described in its original paper to match paired minutiae. As shown in Tab. III, FingerSafe achieves significantly strong protection performance on black-box traditional non-learning models, improving protection capability by large margins relative to the best baseline (up to 76.36% in verification and 49.26% in identification), demonstrating strong transferability even for traditional minutiae or ScatNet features, without the reliance on possibly unstable DNN-extracted features. We also provide an illustration of the extracted ScatNet features and minutiae in Fig. 5. We show that by changing high-level ridge semantics, FingerSafe can also fundamentally perturb the extracted traditional features even when no parameters have been learned in the overall fingerprint recognition process.
TABLE III
EXPERIMENTAL RESULTS FOR TRADITIONAL FEATURE EXTRACTION METHODS. FINGERSAFE IS EFFECTIVE EVEN WHEN THESE WELL-DEFINED FEATURES ARE USED

| Metric | Method       | Clean | PGD | Unlearn. | Fawkes | Lowkey | Malhotra | FingerSafe |
|--------|--------------|-------|-----|----------|--------|--------|----------|------------|
| Ver.   | MHS-Scat-L2  | 76.37 | 72.14 | 74.75   | 57.09  | 70.77  | 67.08    | 9.70       |
| TPR(%) | FingerNet    | 79.85 | 80.10 | 79.35   | 78.23  | 78.61  | 79.35    | 1.87       |
| Iden.  | MHS-Scat-L2  | 83.82 | 66.18 | 70.59   | 39.71  | 61.03  | 33.82    | 3.68       |
| ACC(%) | FingerNet    | 69.85 | 58.82 | 58.09   | 59.56  | 56.62  | 49.26    | 0.0        |

TABLE IV
EER RESULTS OF BASELINE PROTECTION METHODS AND FINGERSAFE ON SIX DIFFERENT TASKS. AGAIN, FINGERSAFE ACHIEVES SIGNIFICANT IMPROVEMENTS IN PROTECTION RESULTS RELATIVE TO THE BASELINES

| Metric | Method       | Clean | PGD | Unlearn. | Fawkes | Lowkey | Malhotra | FingerSafe |
|--------|--------------|-------|-----|----------|--------|--------|----------|------------|
| Ver.   | ResNet50     | 6.59  | 36.75 | 25.81   | 52.99  | 46.70  | 25.81    | 54.91      |
| TPR(%) | InceptionV3  | 5.66  | 22.26 | 14.99   | 31.47  | 34.58  | 14.80    | 44.90      |
| Iden.  | MHS-ResNet50 | 6.09  | 19.03 | 12.44   | 13.93  | 15.30  | 20.83    | 49.69      |
| ACC(%) | HG-ResNet50  | 11.94 | 16.36 | 15.11   | 18.10  | 14.49  | 15.17    | 52.99      |
|        | MHS-Scat-L2  | 16.85 | 26.37 | 21.39   | 27.18  | 23.20  | 27.92    | 41.73      |
|        | FingerNet    | 19.65 | 21.89 | 20.15   | 20.90  | 21.39  | 21.02    | 64.05      |

4) Different Recognition Thresholds: In addition to developing a protective model with a fixed recognition threshold that maximizes the results on clean images, we evaluate whether FingerSafe can also protect privacy under models with uncertain, or varying recognition thresholds. Specifically, we evaluate FingerSafe and all baselines under six typical settings, namely ResNet50, InceptionV3, HG-ResNet50, MHS-ResNet50, MHS-Scat-L2 and FingerNet. For all settings, we take the Equal Error Rate (EER) as the evaluation metric. Results are listed in Table IV. We also plot the Detection Error Tradeoff (DET) curve in Fig. 6.

From the results above, we can observe that the protection of FingerSafe is consistent under different recognition thresholds. Specifically, FingerSafe achieves up to 42.66% higher EER than the best-performing baseline. As indicated by the DET curve, FingerSafe also achieves strong and consistent protection under different thresholds. The strong protection result of FingerSafe indicates that the dissimilarity between protected image and the original image in the feature space is sufficiently large, such that it is effective under a wide range of thresholds.

5) Training Stage Protection: In this part, we demonstrate that FingerSafe can be directly used to protect against training stage attack without any modification. In contrast to testing stage protection, training stage protection aims to release the protected image $x_{adv}$ on social media, so that hackers who train their model $F_2$ on $x_{adv}$ will find low performance when testing on clean images $x$. In particular, we generate all protected images based on the ResNet50 source model and then feed these images to hackers to train their models (with different architectures and different preprocessing methods). The results in Tab.V lead to several conclusions:

Table V
EXPERIMENTAL RESULTS FOR TRAINING STAGE PROTECTION, TESTED ON DIFFERENT ARCHITECTURES AND DIFFERENT PREPROCESSING METHODS. NOTE THAT RESNET50 REPRESENTS A WHITE-BOX ATTACK IN THIS CASE. DESPITE BEING DESIGNED FOR TESTING STAGE PROTECTION, FINGERSAFE ALSO OUTPERFORMS ALL BASELINES IN TRAINING STAGE PROTECTION

| Metric | Method       | Clean | PGD | Unlearn. | Fawkes | Lowkey | Malhotra | FingerSafe |
|--------|--------------|-------|-----|----------|--------|--------|----------|------------|
| Ver.   | ResNet50     | 91.17 | 92.54 | 79.98   | 86.94  | 81.97  | 89.05    | 79.10      |
| TPR(%) | InceptionV3  | 93.28 | 92.41 | 90.35   | 89.68  | 94.78  | 92.29    | 85.32      |
| Iden.  | MHS-ResNet50 | 95.14 | 90.67 | 91.79   | 89.30  | 91.92  | 92.16    | 83.33      |
| ACC(%) | HG-ResNet50  | 88.06 | 89.93 | 85.95   | 84.70  | 87.81  | 86.32    | 80.97      |
|        | ResNet50     | 94.12 | 86.03 | 84.56   | 83.88  | 81.62  | 82.35    | 75.74      |
|        | InceptionV3  | 91.18 | 90.44 | 90.44   | 89.71  | 90.44  | 92.65    | 83.82      |
|        | MHS-ResNet50 | 93.38 | 86.76 | 86.03   | 91.17  | 86.03  | 91.91    | 80.88      |
|        | HG-ResNet50  | 72.79 | 69.18 | 71.32   | 62.50  | 64.71  | 70.59    | 47.79      |

Fig. 6. DET plots of baselines and FingerSafe on six different tasks. The protection capability of FingerSafe consistently exceeds that of the baselines under different thresholds.

1) In terms of training stage protection, FingerSafe achieves the best performance in all black-box settings by large margins (i.e., up to 14.71% protection improvement over the best baseline). Our method even outperforms Unlearnable Examples by large margins, specifically designed for training stage protection, by large margins.

2) FingerSafe is very effective in white-box settings (ResNet50→ResNet50), achieving a protection improvement of 8.14% over the best baseline.

We hypothesize that the improved performance of FingerSafe in training stage protection is due to the fact that our method aims to attack the high-level semantics (i.e., ridge patterns). These features are used as key evidence by
models attempting to learn and make correct predictions. Thus, perturbing high-level semantics could be interpreted as adding a robust "watermark" that provides stable feature collision, making it more difficult for a DNN to separate these features at training time [69].

C. Naturalness of FingerSafe

The quality of the fingerprint privacy protection also depends on the naturalness of the protected images. We therefore conduct human perception studies on one of the most commonly used crowdsourcing platforms. In more details, participants were asked to evaluate the naturalness of social media images protected by PGD, Unlearnable Examples, Fawkes, Lowkey, Malhotra and FingerSafe. Since naturalness cannot be fairly evaluated without protection performance, for each method, we selected four levels of protection strength using four different $\ell_\infty$ constraints {4/255, 8/255, 16/255, 32/255}, resulting in $6 \times 4.24$ images. All ratings were collected using a 7-point Likert scale, ranging from 1 (very low) to 7 (very high). Responses were collected from 55 anonymous participants on WenJuanXing, a platform similar to Amazon Mechanical Turk (AMT). A sample of our questionnaire is presented in Fig. 7. Each participant was required to rate the naturalness of all images, which were presented in a random order. After all results were collected, the naturalness of the image was calculated by averaging the ratings given by participants, also known as the Mean Opinion Score (MOS) [70]. The results are presented in Fig. 8, where Acc refers to identification accuracy. The goal is to achieve better protection capability (we use 1 minus Acc for convenience) and higher naturalness. For comprehensiveness, we also add the result evaluated by objective imperceptibility metrics, such as MSE [71], PSNR [72], SSIM [73] and LPIPS [74], as shown in Fig. 9. Note that in the field of image quality assessment (IQA), human evaluation is commonly referred to as the most reliable and accurate [75], and is thus treated as the golden standard. As a result, we mainly consider human subjective rating results in this paper.

(1) FingerSafe simultaneously provides better protection capability and naturalness results than all baselines. It is worth noting that our FingerSafe outperforms baseline methods by particularly large margins under a high perturbation budget, which suggests strong application potential in practice.

(2) In subjective evaluation, our FingerSafe is the only method to provide comparatively natural performance (2.98) with satisfying protection capability (> 85%), achieving 59.36% improvement compared with Malhotra, the best-performing baseline (2.98 vs 1.87).

(3) The superiority of FingerSafe is consistent in objective evaluation. With a satisfying protection capability (> 85%), our FingerSafe is 1.52 better in MSE (1.31 vs 2.83, lower the better), 2.06 better in PSNR (33.80 vs 31.74, higher the better), 0.05 better in SSIM (0.93 vs 0.88, higher the better) and 0.023 better in LPIPS (0.063 vs 0.086, lower the better). This shows the naturalness of FingerSafe is favored by human subjective evaluations and objective imperceptibility metrics.

(4) State-of-the-art face privacy protection methods, such as Fawkes and Lowkey, produce images with low naturalness regardless of the $\ell_\infty$ constraint. Those methods are thus less recommended for fingerprint privacy protection.

D. Social Media Protection: A Real-World Study

In this section, we further test our FingerSafe in a real-world social media scenario. To resemble images posted in social
media, we manually collect the fingerprint from 100 classes (identities) and obtain a total of 1000 images (each class contains 10 images). For each class, we took 10 photos with diversified background to simulate social media image, from different angles \((-45^\circ, 0^\circ, 45^\circ)\) horizontally, \(-30^\circ, 30^\circ\) vertically), and at different distances \(0.15m, 0.3m\) using an iPhone 12 Pro camera. Sample images are presented in Fig. 10. For each clean image, we first use FingerSafe to generate the protected images based on a ResNet50 source model, then share these images under private mode on Twitter and Facebook; we then download all images and provide an appropriate segmentation [76] for subsequent fingerprint recognition. The overall pipeline is shown in Fig. 11. Note that the upload and download process is nontrivial. While the preprocessing used by Twitter and Facebook is unknown to us, we do find the size of image being compressed after downloading, thus differs from original images we collected.

Specifically, our preprocessing steps mainly follows [76], with procedures detailed as follows. (1) after the image is acquired, we filter the background by first generating a binary mask to segment the image using Otsu’s threshold, suppress noise in the mask via erosion and dilation, and find the largest contour containing fingerprint; (2) we get a fingerprint image by masking the image to filter out the background, locating and segmenting the area containing the fingerprint, correcting the rotation of the fingerprint, and cropping the fingers to the first knuckle; (3) we then use a preprocessing algorithm (i.e., MHS) to extract the ridge features. Note that noises is injected into the masked area only. After noise injection, we record the adversarial noise and add it back to the original position of the first knuckle. In this way, we can simulate the real-world hacking and protection processes, which involve the use of unknown models, preprocessors, etc. on these platforms.

The evaluation results can be found in Tab. VI. Compared with baselines, FingerSafe consistently achieves superior performance on real-world photographic images (27.5% protection improvement over the best baseline). We further emphasize that FingerSafe achieves the best performance when preprocessors are used, which more closely resembles a real-world fingerprint recognition scenario [18].

Moreover, regarding the protection capability in the real world, it is worth noting that our photographs and HKPolyU dataset contains images of different sources, classes, collection methods, etc. Thus, the results conducted on HKPolyU dataset (i.e., Table. I - V) and our collected images (i.e., Table. VI) are not directly comparable. Also note that that for the experiment of our real world images, the hyperparameters \(\lambda\) and \(\gamma\) used for HKPolyU dataset [58] also achieve the best results without additional tuning, showing the set of hyperparameters we use is general, and can be directly applied to images in other domains. See our ablations for further details.

E. Discussion

In this section, we provide further evaluations of FingerSafe, including results under JPEG compression, execution time analysis and comparison with ad-hoc protection methods such as blurring and pixelization.

1) Robustness Under JPEG Compression: Since social media platforms often use JPEG compression when uploading pictures to save space, we here evaluate the performance of FingerSafe under JPEG compression [77] (also known as an adversarial defense method [78]). Specifically, we first generate the protected images on a ResNet50 backbone, then compress the images using JPEG compression at different levels of image quality (lower quality means a higher compression ratio). Finally, we feed the images into another ResNet-50 model using MHS as the preprocessor. The results in Fig. 12 demonstrate that the protection of FingerSafe is robust across all levels of image quality at different JPEG compression rates (the probability of hackers succeeding is at most 1.47%). Moreover, by observing the clean image and baseline trends, we can conclude that the drop in accuracy in baseline methods is probably because of decreased image quality rather than a stronger attacking capability.
methods, in line with existing implementations [42], and \( \alpha = 1e^{-4} \) for Fawkes [9], adhering to their own implementations. The variations in attack objectives are indicated by ↑ or ↓ in Figure 13, with detailed objectives for each method available in the Supplementary Materials.

3) Comparison of Execution Time: Protection in social media must be fast to ensure satisfactory user experience. We report the average execution time of protecting a single image using FingerSafe and baseline methods in Tab. VII. According to Programs Wiki, \(^8\) a response speed of 2-5 seconds is acceptable for users. The execution time of FingerSafe meets this constraint.

A closer look at the results helps us to better analyze attack results of FingerSafe and all baselines. The execution time of FingerSafe is comparable with PGD, Unlearn and Malhotra, since they are all iterative gradient based method, demonstrating its real-time protection capability. The protection of Fawkes relies on C&W style [34] attack, which generally requires larger iterations. LowKey use iterative gradient based method, but simultaneously attacks an ensemble of models and relies on optimizing LPIPS metric [74], introducing more forward and backward propagation that costs additional computation time.

4) The Effect of ad-hoc Protection Methods: We next use experiments to support our claim that ad-hoc protection methods are ineffective. Specifically, we investigate two kind of ad-hoc protection methods, Gaussian blurring and pixelization. For Gaussian blurring, we smooth the fingerprint area using a Gaussian kernel with kernel size of 15 and adjust the strength of the Gaussian blurring by changing the standard deviation \( \sigma \) in the Gaussian kernel. For pixelization, we pixelize different percentages of fingerprints to a 10 × 10 mosaic. Then, we use an off-the-shelf deblurring algorithm, DeblurGAN [80], and the inpainting method Deepfill [81] to reverse the blurring and pixelization process. The protection strength was selected by searching images with similar naturalness to those produced by FingerSafe. For Gaussian protection, we select \( \sigma = \{1.5, 2, 2.5\} \). For pixelization, we select the pixelization percentage \( r = \{30\%, 40\%, 50\%\} \). While pixelizing 30% of the image yields low naturalness, we nonetheless select 30% as the lowest level of blurring, since further decreasing \( r \) leads to negligible protection. The results can be found in Tab. VIII. We draw several conclusions as follows:

1) FingerSafe outperforms ad-hoc protection methods by a large margin in transferability (21.76% improvement over the best-performing method in blurring, 10.69% improvement over the best-performing method in pixelization). Moreover, while providing strong protection, FingerSafe also provides a more natural appearance than all settings of pixelization and

---

\(^8\)https://programs.wiki/wiki/startup-optimization-of-android-performance-optimization.html
TABLE VIII
THE PERFORMANCE OF TWO AD-HOC PROTECTION METHODS (i.e., BLUR AND PIXELIZATION) AFTER SIMPLE DE-NOISING METHODS (i.e., DEBLUR AND INPAINTING). IN THIS TABLE, B(σ) AND P(τ) REPRESENT DIFFERENT STRENGTH OF BLURRING AND PIXELIZATION, UNDER STANDARD Deviation σ AND PIXELIZATION Percentage τ

| Method          | Clean | B(1.5) | B(2) | B(2.5) | B(30%) | B(40%) | B(50%) | FingerSafe |
|-----------------|-------|--------|------|--------|--------|--------|--------|------------|
| ResNet50        | 91.17 | 51.12  | 35.20| 23.51  | 74.25  | 58.96  | 37.06  | 0.00       |
| HG-ResNet50     | 88.06 | 70.77  | 67.04| 62.83  | 68.53  | 61.94  | 51.19  | 4.60       |
| HG-Scat-L2      | 75.75 | 36.97  | 33.33| 27.11  | 29.85  | 24.88  | 16.04  | 5.35       |
| Naturalness     | 6.20  | 4.00   | 3.60 | 3.04   | 2.16   | 2.05   | 1.78   | 3.56       |

TABLE IX
GENERATING FINGERSAFE ON DIFFERENT SOURCE MODELS. THE RESULT OF FINGERSAFE IS INSENSITIVE TO THE CHOICE OF SOURCE MODELS

| Verification TPR (%) | From/To | ResNet50 | HG-ResNet50 | MHS-Scat-L2 |
|----------------------|---------|----------|-------------|-------------|
|                      | ResNet50| 0.00     | 4.60        | 9.70        |
|                      | InceptionV3| 0.12   | 4.10        | 7.96        |
|                      | DenseNet121| 0.00   | 6.72        | 9.70        |

Fig. 14. Ablation study on different λ that controls \( L_\sigma \). Experiments conducted on HKPolyU dataset. Dotted line indicates the average trend of accuracy. FingerSafe yields the best result with \( \lambda \geq 100 \).

F. Ablation Studies

In this section, we conduct ablation studies to verify the effect of different loss terms and hyperparameters, namely \( \lambda \) that controls orientation field distortion loss \( L_\sigma \) and \( \gamma \) that controls local contrast suppression loss \( L_C \).

1) Effect of Different Loss Terms: We conduct ablation studies to better understand the contributions of our two main loss terms, i.e., the orientation field distortion loss and local contrast suppression loss. We argue that the orientation field distortion loss \( L_\sigma \) mainly contributes to the transfer attack in FingerSafe, while the local contrast suppression loss \( L_C \) provides the natural appearance. To test these hypotheses, we conduct an experiment by exploring different loss term combinations. Beginning with a set of fixed \( L_{adv} \), we optimize FingerSafe with the loss functions \( L_\sigma \), \( L_C \) and \( L_\sigma + L_C \) respectively. As shown in Tab. X, the accuracy drops significantly under the \( L_\sigma \) setting (i.e., in MHS-ResNet, 0% under \( L_\sigma \) and 0.74% under \( L_\sigma + L_C \) compared with 25.00% under \( L_C \)). We evaluate naturalness using the same settings in section IV-C. The naturalness is restored by \( L_C \) (i.e., 2.73 under \( L_\sigma \) and 4.75 under \( L_\sigma + L_C \), p<.001, the improvement is significant). These experimental results prove the validity of our claim that \( L_\sigma \) and \( L_C \) achieved their desired goal.

2) The Effect of Hyperparameter \( \lambda \): Hyperparameter \( \lambda \) controls the level of transferability. We evaluate the effectiveness of \( \lambda \) on the ResNet50 backbone and transfer it to different black-box models. We set the value of \( \lambda \) to 1, 10, 100, and 1000, respectively. We first evaluate the result in HKPolyU dataset [58]. As illustrated in Fig. 14, the model accuracy first

We use the nonparametric Wilcoxon signed-rank test to determine whether the improvement of \( L_C \) is at a significant level.
drops sharply with the increase of $\lambda$, then remains relatively stable as $\lambda$ further increases. The decline of model accuracy with preprocessing is more significant, from $71.27\%$ to $4.85\%$ as $\lambda$ increases from 1 to 100. This indicates that perturbing model-shared semantics is a more effective means of attacking black-box preprocessors. When $\lambda$ increases from 100 to 1000, the accuracy increases only slightly, meaning the perturbation induced by $L_O$ has saturated.

Surprisingly, we find that the best $\lambda$ for maximum protection capability in our real world images is identical to the best $\lambda$ of the HKPolyU dataset (i.e., $\lambda = 100$). This is a strong evidence that the hyperparameter $\lambda$ for our orientation field distortion loss can generalize to multiple image domains and protection settings. Specifically, we evaluate the protection result of our real world images on the ResNet50 backbone, transfer the protected images to ResNet50, MHS-ResNet50 and MHS-Scat-L2. We set the value of $\lambda$ to 1, 10, 100, and 1000, respectively. We follow the pipeline in our social media protection settings, which uploads and download all images on Twitter. As shown in Fig. 15, the trend of protection capability follows the trend in the HKPolyU dataset: as $\lambda$ increases from 1 to 100, the protection capability first increases sharply. The average TPR decreases from 74.17% to 32.63%, and average ACC decreases from 60.41% to 31.25%. As $\lambda$ further increases from 100 to 1000, the protection capability gets saturated and does not further increase, resulting in the average TPR decreasing from 32.63% to 30.4% and average ACC decreasing from 31.25% to 30.08%, respectively. To sum up, while our experiments on real world images directly follow the $\lambda$ of HKPolyU without change, we surprisingly find that it is also the best hyperparameter for real world images, which implies the magnitude for our orientation field distortion loss is suitable for multiple image domains.

3) The Effect of Different $\gamma$: Hyperparameter $\gamma$ controls the level of naturalness. We evaluate the effectiveness of $\gamma$ on the ResNet50 backbone and transfer it to different black-box models. We set the value of $\gamma$ to 5, $5 \times 10^2$, $5 \times 10^4$, and $5 \times 10^6$, respectively. We first evaluate the result in HKPolyU dataset. As illustrated in Fig. 16, the model accuracy first increases slightly with the increase of $\gamma$ (e.g., from 2.61% TPR to 4.85% TPR when $\gamma$ changes from 5 to $5 \times 10^2$), then increases substantially as $\gamma$ further increases (e.g., from 4.85% TPR to 41.26% TPR when $\gamma$ changes from $5 \times 10^2$ to $5 \times 10^4$). When $\gamma = 5 \times 10^6$, all protective patterns generated by the orientation distortion loss $L_O$ are simply wiped out. Due to space limit, we show the overall trend of the effect of different $\gamma$ here, and defer a closer look at $\gamma$ in range $5 \times 10^2$ to $5 \times 10^4$ to supplementary materials. We find the protection capability of FingerSafe is stable with $\gamma < 5 \times 10^3$, while achieving best protection-naturalness tradeoff when $\gamma = 5 \times 10^2$.

The best $\gamma$ for our real world images is also identical to what we used in HKPolyU dataset (i.e., $\gamma = 5 \times 10^2$). Identical to the settings above, we evaluate the effectiveness of $\gamma$ on the ResNet50 backbone and transfer it to different black-box models, with $\gamma$ set to 5, $5 \times 10^2$, $5 \times 10^4$, and $5 \times 10^6$, respectively. The results are shown in Fig. 17. Specifically, the protection capability stays similar when $\gamma$ increases from 5 to $5 \times 10^2$ (e.g., from 31.43% average TPR to 32.63% average TPR when $\gamma$ changes from 5 to $5 \times 10^2$), with naturalness increasing largely. However, when $\gamma$ further increases from $5 \times 10^2$ to $5 \times 10^6$, the improvement of naturalness comes at a cost of protection capability (e.g., from 32.63% average TPR to 77.48% average TPR when $\gamma$ changes from 5 to $5 \times 10^2$). Due to space limitations, we show the overall trend of the effect of different $\gamma$ here, and defer a closer look at $\gamma$ in range $5 \times 10^2$ to $5 \times 10^6$ to supplementary materials. We find the protection capability of FingerSafe in real world is stable with $\gamma < 5 \times 10^3$, while we select $\gamma = 5 \times 10^2$ for best protection-naturalness tradeoff.

V. CONCLUSION

Biometric information (e.g., fingerprints) can be easily stolen from the social media images, with serious and irreversible consequences for personal security. Countermeasures have emerged as a new way to guard against fingerprint leakage. However, existing works are either weak in black-box transferability or produce images with an unnatural
appearance. To tackle this challenge, this paper proposes a hierarchical attack framework named FingerSafe that works by leveraging the low-to-high hierarchy of perception, in that it both attacks the high-level semantics and suppresses low-level stimuli. Extensive experiments in both digital and real-world scenarios demonstrate that FingerSafe outperforms other methods by large margins.

In the future, we will continue to improve our FingerSafe tool for fingerprint privacy protection, specifically by making the recognition and processing speed more suitable for real-world application. Moreover, we hope to collaborate with the industry to further implement automatic fingerprint protection in social media applications, preferably using a fast and highly optimized engineering method. Finally, we wish to collaborate with social media industries in order to apply FingerSafe to protect fingerprint privacy leakage in real-world applications.

REFERENCES

[1] A. Malhotra, S. Chhabra, M. Vatsa, and R. Singh, “On privacy preserving anonymization of fingerprint-selfies,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 120–128.

[2] Y. Zhang, Z. Chen, H. Xue, and T. Wei, “Fingerprints on mobile devices: Abusing and leaking,” in Proc. Black Hat Conf., 2015.

[3] K. Cao and A. K. Jain, “Hacking mobile phones using 2D printed fingerprints,” Michigan State Univ., East Lansing, MI, USA, Tech. Rep. MSU-CSE-16-2, 2016.

[4] P. Rascagneres and V. Ventura, “Fingerprint cloning: Myth or reality?” Cisco Talos, Tech. Rep., Apr. 2020, vol. 8.

[5] R. Gross, L. Sweeney, F. de la Torre, and S. Baker, “Model-based face de-identification,” in Proc. Conf. Comput. Vis. Pattern Recognit. Workshop, 2006, p. 161.

[6] C. Szegedy et al., “Intriguing properties of neural networks,” 2013, arXiv:1312.6104.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, arXiv:1412.6572.

[8] H. Huang, X. Ma, S. Monazam Erfani, J. Bailey, and Y. Wang, “Unlearnable examples: Making personal data unexploitable,” 2021, arXiv:2101.04898.

[9] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, “Fawkes: Preventing privacy attacks against unauthorized deep learning models,” in Proc. 29th USENIX Secur. Symp., 2020, pp. 1589–1604.

[10] X. Yang et al., “Towards face encryption by generating adversarial identity masks,” 2020, arXiv:2003.06814.

[11] V. Cherepanova et al., “LowKey: Leveraging adversarial attacks to protect social media users from facial recognition,” 2021, arXiv:2101.07922.

[12] I. J. A. Groen, E. H. Silson, and C. I. Baker, “Contributions of low-and-high-level properties to neural processing of visual scenes in the human brain,” Phil. Trans. Roy. Soc. B, Biol. Sci., vol. 372, no. 1714, Feb. 2017, Art. no. 20160102.

[13] J. Wang, A. Liu, Z. Yin, S. Liu, S. Tang, and X. Liu, “Dual attention suppression attack: Generate adversarial camouflage in physical world,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 8561–8570.

[14] M. G. Berman et al., “The perception of naturalness correlates with low-level visual features of environmental scenes,” PLoS ONE, vol. 9, no. 12, Dec. 2014, Art. no. e114572.

[15] L. Hong, Y. Wan, and A. Jain, “Fingerprint image enhancement: Algorithm and performance evaluation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 8, pp. 777–789, Jul. 1998.

[16] Y. Tadmor and D. I. Tolhurst, “Calculating the contrasts that retinal ganglion cells and LGN neurones encounter in natural scenes,” Vis. Res., vol. 40, no. 22, pp. 3145–3157, Oct. 2000.

[17] A. Jain, L. Hong, and R. Bolle, “On-line fingerprint verification,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 4, pp. 302–314, Apr. 1997.

[18] A. Sankaran, A. Malhotra, A. Mittal, M. Vatsa, and R. Singh, “On smartphone camera based fingerprint authentication,” in Proc. IEEE 7th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS), Sep. 2015, pp. 1–7.

[19] C. Lin and A. Kumar, “Multi-siamese networks to accurately match contactless to contact-based fingerprint images,” in Proc. IEEE Int. Joint Conf. Biometrics (JCB), Oct. 2017, pp. 277–285.

[20] F. Afzar, M. Arif, and M. Hussain, “Fingerprint identification and verification system using minutiae matching,” in Proc. Nat. Conf. Emerg. Technol., vol. 2, 2004, pp. 141–146.

[21] R. Cappelli, M. Ferrara, and D. Maltoni, “Minutia cylinder-code: A new representation and matching technique for fingerprint recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 12, pp. 2128–2141, Dec. 2010.

[22] Q. Zheng, A. Kumar, and G. Pan, “Suspecting less and doing better: New insights on palmprint identification for faster and more accurate matching,” IEEE Trans. Inf. Forensics Security, vol. 11, no. 3, pp. 633–641, Mar. 2016.

[23] A. Malhotra, A. Sankaran, M. Vatsa, and R. Singh, “On matching fingerselies using deep scattering networks,” IEEE Trans. Biometrics, Behav. Identity Sci., vol. 2, no. 4, pp. 350–362, Oct. 2020.

[24] J. Bruna and S. Mallat, “Invariant scattering convolution networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1872–1886, Aug. 2013.

[25] Y. Tang, F. Gao, J. Feng, and Y. Liu, “FingerNet: An unified deep network for fingerprint minutiae extraction,” in Proc. IEEE Int. Joint Conf. Biometrics (IJCB), Sep. 2017, pp. 22–30.

[26] C. Lin and A. Kumar, “Contactless and partial 3D fingerprint recognition using multi-view deep representation,” Pattern Recognit., vol. 83, pp. 314–327, Nov. 2018.

[27] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar, “Blind authentication: A secure crypto-biometric verification protocol,” IEEE Trans. Inf. Forensics Security, vol. 5, no. 2, pp. 255–268, Jun. 2010.

[28] Y. Sun, Deep Learning Face Representation by Joint Identification-Verification, Cambridge, MA, USA: MIT Press, 2015.

[29] H. Huang, X. Ma, S. Monazam Erfani, J. Bailey, and Y. Wang, “Unlearnable examples: Making personal data unexploitable,” 2021, arXiv:2101.04898.

[30] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, “Fawkes: Preventing privacy attacks against unauthorized deep learning models,” in Proc. 29th USENIX Secur. Symp., 2020, pp. 1589–1604.

[31] X. Yang et al., “Towards face encryption by generating adversarial identity masks,” 2020, arXiv:2003.06814.

[32] V. Cherepanova et al., “LowKey: Leveraging adversarial attacks to protect social media users from facial recognition,” 2021, arXiv:2101.07922.

[33] J. Wang, A. Liu, Z. Yin, S. Liu, S. Tang, and X. Liu, “Dual attention suppression attack: Generate adversarial camouflage in physical world,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 8561–8570.

[34] M. G. Berman et al., “The perception of naturalness correlates with low-level visual features of environmental scenes,” PLoS ONE, vol. 9, no. 12, Dec. 2014, Art. no. e114572.

[35] L. Hong, Y. Wan, and A. Jain, “Fingerprint image enhancement: Algorithm and performance evaluation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 8, pp. 777–789, Jul. 1998.

[36] Y. Tadmor and D. I. Tolhurst, “Calculating the contrasts that retinal ganglion cells and LGN neurones encounter in natural scenes,” Vis. Res., vol. 40, no. 22, pp. 3145–3157, Oct. 2000.

[37] A. Jain, L. Hong, and R. Bolle, “On-line fingerprint verification,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 4, pp. 302–314, Apr. 1997.

[38] A. Sankaran, A. Malhotra, A. Mittal, M. Vatsa, and R. Singh, “On smartphone camera based fingerprint authentication,” in Proc. IEEE 7th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS), Sep. 2015, pp. 1–7.
[44] M. J. Wilber, V. Shmatikov, and S. Belongie, “Can we still avoid automatic face detection?” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2016, pp. 1–9.

[45] R. McPherson, R. Shokri, and V. Shmatikov, “Defeating image obfuscation with deep learning,” 2016, arXiv:1609.00408.

[46] S. J. Oh, R. Benenson, M. Fritz, and B. Schiele, “Faceless person recognition: Privacy implications in social media,” in Proc. ECCV. Cham, Switzerland: Springer, 2016, pp. 19–35.

[47] K. Nandakumar and A. K. Jain, “Biometric template protection: Bridging the performance gap between theory and practice,” IEEE Signal Process. Mag., vol. 32, no. 5, pp. 88–100, Sep. 2015.

[48] A. Sarkar and B. K. Singh, “A review on performance, security and various biometric template protection schemes for biometric authentication systems,” Multimedia Tools Appl., vol. 79, nos. 37–38, pp. 27721–27776, Oct. 2020.

[49] P. Li, J. Tian, X. Yang, P. Shi, and Y.-Y. Zhang, “Biometric template protection,” J. Softw., vol. 20, no. 6, pp. 1553–1573, Jul. 2009.

[50] J. B. Kho, J. Kim, I.-J. Kim, and A. B. J. Teoh, “Cancelable fingerprint template design with randomized non-negative least squares,” Pattern Recognit., vol. 91, pp. 245–260, Jul. 2019.

[51] W. Yang, S. Wang, J. Hu, G. Zheng, and C. Vaili, “Security and accuracy of fingerprint-based biometrics: A review,” Symmetry, vol. 11, no. 2, p. 141, Jan. 2019.

[52] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” 2013, arXiv:1312.6034.

[53] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, “Understanding neural networks through deep visualization,” 2015, arXiv:1506.06579.

[54] S. Minaee, A. Abdollahshidi, H. Su, M. Bennamoun, and D. Zhang, “Biometrics recognition using deep learning: A survey,” 2019, arXiv:1912.00271.

[55] R. Veale, Z. M. Hafed, and M. Yoshida, “How is visual salience computed in the brain? Insights from behaviour, neurobiology and modelling,” Phil. Trans. Roy. Soc. B, Biol. Sci., vol. 372, no. 1714, Feb. 2017, Art. no. 20160113.

[56] E. S. Levitin and B. T. Polyak, “Constrained minimization methods,” USSR Comput. Math. Math. Phys., vol. 6, no. 5, pp. 1–50, 1966.

[57] A. Virmaux and K. Scaman, “Lipschitz regularity of deep neural networks: Analysis and efficient estimation,” in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 3839–3848.

[58] C. Lin and A. Kumar, “Matching contactless and contact-based conventional fingerprint images for biometrics identification,” IEEE Trans. Image Process., vol. 27, no. 4, pp. 2008–2021, Apr. 2018.

[59] P. Lu, J. Yu, X. Peng, Z. Zhao, and X. Wang, “Gray2ColorNet: Transfer more colors from reference image,” in Proc. 28th ACM Int. Conf. Multimedia, Oct. 2020, pp. 3210–3218.

[60] S. Chopra, A. Malhotra, M. Vatsa, and R. Singh, “Unconstrained fingerphoto database,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2018, pp. 1–12.

[61] N. Kaur, “A study of biometric identification and verification system,” in Proc. IEEE Int. Conf. Advance Comput. Innov. Technol. Eng. (ICACITE), Mar. 2021, pp. 60–64.

[62] D. Peralta et al., “A survey on fingerprint minutiae-based local matching for verification and identification: Taxonomy and experimental evaluation,” Inf. Sci., vol. 315, pp. 67–87, Sep. 2015.

[63] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[64] C. Szegedy, V. Vanhoucke, S. Ioffie, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

[65] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2261–2269.

[66] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” in Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent. Cham, Switzerland: Springer, 1998.

[67] K. Hill, “The secretive company that might end privacy as we know it,” in Ethics of Data and Analytics. New York, NY, USA: Auerbach, 2022, pp. 170–177.

[68] A. Paszke, “Automatic differentiation in PyTorch,” Tech. Rep., 2017.

[69] A. Shafahi et al., “Poison frogs! Targeted clean-label poisoning attacks on neural networks,” 2018, arXiv:1804.00792.

[70] D. Ghadiyaram and A. C. Bovik, “Massive online crowdsourced study of subjective and objective picture quality,” IEEE Trans. Image Process., vol. 25, no. 1, pp. 372–387, 2015.

[71] U. Sara, M. Akter, and M. S. Uddin, “Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study,” J. Comput. Commun., vol. 7, no. 3, pp. 8–18, 2019.

[72] A. Hore and D. Ziou, “Image quality metrics: PSNR vs. SSIM,” in Proc. 20th Int. Conf. Pattern Recognit., Aug. 2010, pp. 2368–2369.

[73] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[74] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 586–595.

[75] G. Zhai and X. Min, “Perceptual image quality assessment: A survey,” Sci. China Inf. Sci., vol. 63, no. 11, pp. 1–52, Nov. 2020.

[76] J. Priesnitz, R. Huesmann, C. Rathgeb, N. Buchmann, and C. Busch, “Mobile touchless fingerprint recognition: Implementation, performance and usability aspects,” 2021, arXiv:2103.03038.

[77] G. K. Wallace, “The JPEG still picture compression standard,” Commun. ACM, vol. 34, no. 4, pp. 30–44, Apr. 1991.

[78] N. Das et al., “Keeping the bad guys out: Protecting and vaccinating deep learning with JPEG compression,” 2017, arXiv:1705.02900.

[79] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[80] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, “DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Oct. 2019, pp. 8878–8887.

[81] I. Shiri, P. Sheikhzadeh, and M. R. Ay, “Deep-fill: Deep learning based sinogram domain gap filling in positron emission tomography,” 2019, arXiv:1906.07168.