Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition

Mahmood Sharif
Carnegie Mellon University
Pittsburgh, PA, USA
mahmoods@cmu.edu

Sruti Bhagavatula
Carnegie Mellon University
Pittsburgh, PA, USA
srutib@cmu.edu

Lujo Bauer
Carnegie Mellon University
Pittsburgh, PA, USA
lbauer@cmu.edu

Michael K. Reiter
University of North Carolina
Chapel Hill, NC, USA
reiter@cs.unc.edu

ABSTRACT

Machine learning is enabling a myriad innovations, including new algorithms for cancer diagnosis and self-driving cars. The broad use of machine learning makes it important to understand the extent to which machine-learning algorithms are subject to attack, particularly when used in applications where physical security or safety is at risk.

In this paper, we focus on facial biometric systems, which are widely used in surveillance and access control. We define and investigate a novel class of attacks: attacks that are *physically realizable* and *inconspicuous*, and allow an attacker to evade recognition or impersonate another individual. We develop a systematic method to automatically generate such attacks, which are realized through printing a pair of eyeglass frames. When worn by the attacker whose image is supplied to a state-of-the-art face-recognition algorithm, the eyeglasses allow her to evade being recognized or to impersonate another individual. Our investigation focuses on white-box face-recognition systems, but we also demonstrate how similar techniques can be used in black-box scenarios, as well as to avoid face *detection*.

1. INTRODUCTION

Machine learning (ML) is a technology that is profoundly changing the world. Some of the transformative innovations that it enables, such as customized search results and automated translations, might seem minor. Other innovations are clearly revolutionary, enabling new applications or significantly increasing our quality of life—examples include algorithms for cancer diagnosis and self-driving vehicles.

The broad use of ML has also caused a rising interest to understand the extent to which ML algorithms and applications are vulnerable to attacks. These attacks are especially worrisome due to humans' increasing reliance on technology and ML (sometime trusting them even with their own lives [36]). The domains that received the majority of the attention have a common characteristic: the adversary is able to precisely control the digital representation of the input to the ML tools. For instance, in spam detection, the attacker can control the contents of the email and the address of the sender. Hence, unsurprisingly, it has been shown that attacks in many of these domains can be effective at evading ML classifiers designed to detect them [23, 35, 43].

In this paper, we explore the extent to which facial biometric systems are vulnerable to attacks. These systems are widely used for various sensitive purposes, including surveillance and access control [26, 27, 28]. Thus, attackers who mislead them can cause severe ramifications.

In contrast to domains previously explored, attackers who aim to mislead facial biometric systems often do not have precise control over the systems' input. Rather, attackers may be able to control only their (physical) appearance. The process of converting the physical scene into a digital input is not under the attackers' control, and is additionally affected by factors such as lighting conditions, pose, and distance. As a result, it is more difficult for attackers to manipulate or craft inputs that would cause misclassification than it would be, for example, in the domain of spam detection.

Another difficulty that attackers face in the case of facial biometric systems is that manipulating inputs to evade the ML classifiers might be easily observed from outside the systems. For example, attackers can wear an excessive amount of makeup in order to evade a surveillance system deployed at banks [14]. However, these attackers may draw an increased attention from bystanders, and can be deterred by traditional means, e.g., the police.

In the light of these two challenges, we define and study a new class of attacks: attacks that are *physically realizable* and at the same time are *inconspicuous*. In such attacks, the attacker manipulates the physical state that an ML algorithm is analyzing rather than the digitized representation of this state. At the same time, the manipulations of physical state needed by the attacks are sufficiently subtle that they are either imperceptible to humans or, if perceptible, seem natural and not representative of an attack.

Inconspicuousness We focus, unlike most related work (e.g., [7]), on attacks that are *inconspicuous*, i.e., a person who is physically present at a scene, or a person who looks at

CCS'16 October 24-28, 2016, Vienna, Austria

ACM ISBN 978-1-4503-4139-4/16/10.

DOI: http://dx.doi.org/10.1145/2976749.2978392

the input gathered by a sensor (e.g., by watching a camera feed), should not notice that an attack is taking place.

We believe that this focus on attacks that are not readily apparent to humans is important for two reasons: First, such attacks can be particularly pernicious, since they will be resistant to at least cursory investigation. Hence, they are a particularly important type of attacks to investigate and learn how to defend against. Second, such attacks help the perpetrators (whether malicious or benign) achieve *plausible deniability*; e.g., a person seeking merely to protect her privacy against aggressive use of face recognition by merchants will plausibly be able to claim that any failure of an ML algorithm to recognize her is due to error or chance rather than deliberate subterfuge.

Physical Realizability In this work, we are interested in attacks that can be *physically implemented* for the purpose of fooling facial biometric systems. In addition, we focus on attacks on state-of-the-art algorithms. Previous work on misleading ML algorithms often targets algorithms that are not as sophisticated or as well trained as the ones used in practice (see Sec. 2), leaving the real-world implications of possible attacks unclear. In contrast, we focus on those attacks that pose a realistic and practical threat to systems that already are or can easily be deployed.

We divide the attacks we study into two categories, which differ both in the specific motivations of potential attackers and in the technical approaches for implementing the attacks. The two categories of attacks we are interested in are *dodging and impersonation*. Both dodging and impersonation target face recognition systems (FRSs) that perform multi-class classification—in particular, they attempt to find the person to whom a given face image belongs.

Impersonation In an impersonation attack, the adversary seeks to have a face recognized as *a specific other face*. For example, an adversary may try to (inconspicuously) disguise her face to be recognized as an authorized user of a laptop or phone that authenticates users via face recognition. Or, an attacker could attempt to confuse law enforcement by simultaneously tricking multiple geographically distant surveillance systems into "detecting" her presence in different locations.

Dodging In a dodging attack, the attacker seeks to have her face misidentified as *any* other arbitrary face. From a technical standpoint, this category of attack is interesting because causing an ML system to arbitrarily misidentify a face should be easier to accomplish with minimal modifications to the face, in comparison to misclassifying a face as a particular impersonation target. In addition to malicious purposes, dodging attacks could be used by benign individuals to protect their privacy against excessive surveillance.

In this paper we demonstrate inconspicuous and physically realizable dodging and impersonation attacks against FRSs. We detail the design of eyeglass frames that, when printed and worn, permitted three subjects (specifically, the first three authors) to succeed at least 80% of the time when attempting dodging against state-of-the-art FRS models. Other versions of eyeglass frames allowed subjects to impersonate randomly chosen targets. For example, they allowed one subject, a white male, to impersonate Milla Jovovich, a white female, 87.87% of the time; a South-Asian female to impersonate a Middle-Eastern male 88% of the time; and a Middle-Eastern male to impersonate Clive Owen, a white male, 16.13% of the time. We also show various other results as extensions to our core methods, in accomplishing impersonation with less information (so called *black-box* models, see Sec. 3) and in evading face *detection*.

Our contributions in this paper are threefold:

I. We show how an attacker that knows the internals of a state-of-the-art FRS can physically realize impersonation and dodging attacks. We further show how these attacks can be constrained to increase their inconspicuousness (Sec. 4–5).

II. Using a commercial FRS [25], we demonstrate how an attacker that is unaware of the system's internals is able to achieve inconspicuous impersonation (Sec. 6).

III. We show how an attacker can be invisible to facial biometric systems by avoiding detection through the Viola-Jones face detector, the most popular face-detection algorithm [42] (Sec. 7).

Sec. 2–3 survey related work and provide background. In Sec. 4 we describe our method for generating physically realizable white-box attacks, and detail experiments that show its effectiveness in Sec. 5. We demonstrate that similar attacks can be carried out against black-box FRSs in Sec. 6, and that they can additionally be used to evade face *detection* in Sec. 7. We discuss the implications and limitations of our approach in Sec. 8 and conclude with Sec. 9.

2. RELATED WORK

Face Detection Systems (FDSs) and FRSs are traditionally evaluated with the implicit assumption of zero-effort attackers (i.e., the attackers do not actively attempt to fool the system) [33, 42]. Nevertheless, there is extensive literature on attacks targeting FDSs and FRSs. In this section we review the most noteworthy attempts. Broadly, however, existing work is different than ours in at least one of the following ways: it does not attempt to achieve inconspicuousness and physical realizability; it tries to achieve different goals than ours (e.g., compromising the privacy of the enrolled users); or the attacks target algorithms that are not as advanced as the state of the art.

Face Detection Due to its high accuracy and speed, the object-detection algorithm proposed by Viola and Jones [42] (VJ) has become widely used for general object detection, and is by far the dominant face detector used in practice [19, 26, 30]. Motivated by increasing privacy through invisibility to FRSs, Harvey attempted to design makeup and hair styles that can be used to evade detection by the VJ face detector [14]. His approach was based on manually observing how the detector operates, and did not yield inconspicuous results. With similar goals, Yamada et al. leveraged the fact that camera sensors are sensitive to near infra-red light to design light-emitting glasses that can help evade face detection by adding significant noise to the captured image [44]. While effective for evading detection, this technique is neither inconspicuous nor deniable.

Face Recognition Previous work found that subjects can be effectively impersonated to FRSs using 3d-printed masks or face images downloaded from online social networks [7, 22]. These attacks are not inconspicuous, and can largely be thwarted by anti-spoofing mechanisms, such as liveness detection [7, 22].

Other work showed how to compromise the privacy of users enrolled in FRSs [11, 12]. For example, Fredrikson et al. presented *model inversion* attacks, in which face images of enrolled users are reconstructed from neural networks trained to perform face recognition [11]. The goal of these attacks is different than ours. While compromised images can be used to impersonate users, these impersonations will not be inconspicuous, and can be detected via anti-spoofing mechanisms.

Feng and Prabhakaran proposed a system to help artists come up with makeup and hair designs that can be used to prevent FRSs (including Eigenfaces [40] and Fisherfaces [1]) from recognizing subjects [10]. In their approach, they split the face image into seven components, and determined which components deviated most from the general population's average. They hypothesized that these discriminative components are the most significant for distinguishing one person from others, and showed that occluding, painting, or transforming the discriminative components could prevent automated recognition. Their work is different than what is being proposed here in three main ways: First, inconspicuousness and deniability of the attack were not part of their goals. Second, their attack does not enable the attacker to intentionally masquerade as a specific subject in the system. Finally, the algorithms on which they test their attack are not considered the state-of-the-art in face recognition.

Speech Recognition Carlini et al. recently demonstrated physically realizable attacks on speech-recognition systems [5]. They showed how to craft sounds that are difficult or impossible for humans to understand, but are interpreted as specific commands by speech-recognition systems. In contrast to their work, we focus on systems that receive visual inputs through cameras.

3. BACKGROUND

3.1 Threat Model

We assume an attacker who gains access to the FRS to mount a dodging or impersonation attack after the system has been trained. That is, the adversary cannot "poison" the FRS by altering training data, injecting mislabeled data, etc. Rather, our adversary can alter only the composition of inputs to be classified based on her knowledge of the classification model, potentially derived by examining the entire model or by observing its outputs for some inputs.

As Papernot et al. discuss, adversaries differ in the amount of knowledge they have about the system they are attacking [31]. We typically assume that an adversary knows the algorithm used for classification. On the other hand, different assumptions can plausibly be made about the adversary's knowledge of the feature space and the trained classification model. In this paper, we focus on adversaries who know the feature space, since features used for image classification are usually considered to be publicly known [11]. We also assume a *white-box* scenario: the attacker knows the internals (architecture and parameters) of the system being attacked [31]. We do, however, discuss extensions to the black-box scenario in Sec. 6.

3.2 Deceiving Neural Networks

In this paper we focus our exploration on FRSs based on neural networks (NNs) and particularly deep neural networks (DNNs) [16]. DNNs can generalize extremely well in the presence of large training sets and can achieve state-of-the-art results on many challenging ML problems. For example, DNNs were able to outperform humans in the face-verification task—determining whether pairs of face images belong to the same person [17].

Given their high generalization capability, Szegedy et al. surprised the research community by finding that DNNs can be misled by mildly perturbing inputs [39]. More specifically, their work showed that adversarial examples that appear visually indistinguishable from non-adversarial examples can be systemically found under the white-box scenario.

For an input x that is classified as $f(x)$ by the DNN, Szegedy et al.'s goal was to find a perturbation r of minimal norm (i.e., as imperceptible as possible) such that $x + r$ would be classified into a desired class c_t. The objective of finding the perturbation is modeled by:

$$\underset{r}{\operatorname{argmin}} \Big(|f(x+r) - h_t| + \kappa |r| \Big)$$

constrained on $x + r \in [0, 1]$, where f produces a probability distribution (an element of $[0, 1]^N$) over the N possible classes, κ is a constant that can be tuned to balance the misclassification and imperceptibility objectives, h_t is the one-hot vector of class c_t, and $|\cdot|$ is a norm function. Minimizing $|f(x + r) - h_t|$ results in misclassification, whereas minimizing $|r|$ increases the imperceptibility of the perturbation. When $f(\cdot)$ and $|\cdot|$ are differentiable, this optimization problem can be solved via first- or second-order optimization algorithms, such as Gradient Descent [4] or Limited-memory BFGS [29]. For DNNs, $f(\cdot)$ is differentiable, as this is necessary for training the network via back-propagation. The norm function usually used is the Euclidean norm, which is also differentiable. Subsequent work proposed conceptually similar algorithms for generating imperceptible adversarial examples [13, 31].

Initially, the existence of adversarial examples was attributed to the complexity of learning the high-dimensional manifold necessary to separate the classes: it was conjectured that adversarial examples are improbable to encounter in training and hence remain as inevitable "blind spots" when learning from finite, although large, training sets [39]. This conjecture was supported by showing that injecting correctly labeled adversarial examples in the training phase slightly increases the norm of the adversarial perturbations needed to mislead a trained classifier [39]. Later work attributes adversarial examples to the non-flexibility of the classification models [9, 13]. According to this work, since each DNN unit (i.e., neuron) is activated on a linear combination of its inputs, slight changes to each input accumulate to large changes in the unit's output, which results in misclassification.

Previous work highlights an important difference between human and machine vision: while humans are unaffected by small changes to images, machines can be greatly affected. Our research exploits this insight to fool FRSs with adaptations to attacks that are both inconspicuous and physically realizable.

4. TECHNICAL APPROACH

In this section we describe our approach to attacking white-box FRSs. First, we provide the details of the state-of-the-art FRSs we attack (Sec. 4.1). Then, we present how we

build on previous work [13, 31, 39] to formalize how an attacker can achieve impersonation, and how to generalize the approach for dodging (Sec. 4.2). Finally, we discuss how the attacker can conceptually and formally tweak her objective to enable the physical realizability of the attack (Sec. 4.3).

4.1 White-box DNNs for Face Recognition

Using ML models and particularly DNNs, researchers have developed FRSs that can outperform humans in their ability to recognize faces [17]. Parkhi et al. recently developed a 39-layer DNN for face recognition and verification that achieves state-of-the-art performance (and outperforms humans) [33] on the Labeled Faces in the Wild (LFW) [17] challenge, a benchmark for testing FRSs' ability to classify images taken under unconstrained conditions.

To demonstrate dodging and impersonation we use three DNNs: First, we use the DNN developed by Parkhi et al.[1], which we refer to as DNN_A. Second, we trained two DNNs (termed DNN_B and DNN_C) based on the structure of Parkhi et al.'s to recognize celebrities as well as people who were available to us in person for testing real attacks. The purpose of using DNN_B and DNN_C is to test the physically realized versions of our attacks, which require people to wear glasses designed by our algorithm. Testing dodging requires the subjects to be known to the classifier (otherwise they would always be misclassified). Similarly, testing impersonation is more realistic (and more difficult for the attacker) if the DNN is able to recognize the impersonator. The DNNs we use in this work can be conceptualized as differentiable functions that map input images to probability distributions over classes. An image is counted as belonging to the class that receives the highest probability, optionally only if that probability exceeds a predefined threshold.

DNN_A Parkhi et al. trained DNN_A to recognize 2622 celebrities. They used roughly 1000 images per celebrity for training, for a total of about 2.6M images. Even though they used much less data for training than previous work (e.g., Google used 200M [38]), their DNN still achieves 98.95% accuracy, comparable to other state-of-the-art DNNs [17].

DNN_B* and *DNN_C Using DNN_A for physical realizability experiments is not ideal, as it was not trained to recognize people available to us for testing physically realized attacks. Therefore, we trained two additional DNNs.

DNN_B was trained to recognize ten subjects: five people from our lab (the first three authors and two additional researchers who volunteered images for training), and five celebrities for whom we picked images from the PubFig image dataset [21]. In total, the training set contained five females and five males of ages 20 to 53 years. The celebrities we used for training were: Aaron Eckhart, Brad Pitt, Clive Owen, Drew Barrymore, and Milla Jovovich.

DNN_C was trained to recognize a larger set of people, arguably posing a tougher challenge for attackers attempting impersonation attacks. In total, DNN_C was trained to recognize 143 subjects: 140 celebrities from PubFig's [21] evaluation set, and the first three authors.

We trained DNN_B and DNN_C via *transfer learning*, a traditional procedure to train DNNs from other, pre-trained, DNNs [45]. Transfer learning reduces the need for large amounts of data for training a DNN by repurposing a pre-trained DNN for a different, but related, classification task. This is performed by copying an initial set of layers from the existing DNN, appending new layers to them, and training the parameters of the new layers for the new task. Consequently, the layers copied from the old network serve for feature extraction and the extracted features are fed to the new layers for classification. Previous work has shown that transfer learning is effective in training high-performance DNNs from ones that have already been trained when they perform closely related tasks [45].

We followed the suggestion of Yosinski et al. [45] to train DNN_B and DNN_C. More specifically, we used the first 37 layers of DNN_A for feature extraction. Then, we appended a fully connected layer of neurons with a sigmoid activation function followed by a *softmax* layer, and updated the weights in the fully connected layer via the back-propagation algorithm. The neurons in the fully-connected layers serve as linear classifiers that classify the features extracted by DNN_A into identities, and the *softmax* layer transforms their output into a probability distribution. Training more layers (i.e., more parameters to tune) was prohibitive, as the amount of data we could collect from people available for testing physically realizable attacks was limited. We used about 40 images per subject for training—an amount of images that was small enough to collect, but high enough to train highly performing DNNs. On images held aside for testing, DNN_B achieved classification accuracy of 97.43%, and DNN_C achieved accuracy of 96.75%.

Similarly to Parkhi et al., we used 2d affine alignment to align face images to a canonical pose at the input of the DNNs [33]. Additionally, we resized input images to 224×224 (the input dimensionality of the DNNs we use). We used MatConvNet, an NN toolbox for MATLAB, to train DNN_B and DNN_C, run the DNNs, and test our attacks [41].

4.2 Attacking White-box FRSs

Following Parkhi et al. we adopt the *softmaxloss* score to measure the correctness of classifications [33]. Formally, given an input x of class c_x that is classified as $f(x)$ (a vector of probabilities), *softmaxloss* is defined as follows:

$$softmaxloss(f(x), c_x) = -\log\left(\frac{e^{\langle h_{c_x}, f(x)\rangle}}{\sum_{c=1}^{N} e^{\langle h_c, f(x)\rangle}}\right)$$

where $\langle\cdot,\cdot\rangle$ denotes inner product between two vectors, N is the number of classes, and (as discussed in Section 3.2) h_c is the one-hot vector of class c. It follows that *softmaxloss* is lower when the DNN classifies x correctly, and higher when the classification is incorrect. We use *softmaxloss* to define the impersonation and dodging objectives of attackers.

Impersonation An attacker who wishes to impersonate a target t needs to find how to perturb the input x via an addition r to maximize the probability of class c_t. Szegedy et al. defined this as minimizing the distance between $f(x+r)$ and h_t (see Sec. 3.2). Similarly, we define the optimization problem to be solved for impersonation as:

$$\underset{r}{\operatorname{argmin}}\ softmaxloss(f(x+r), c_t)$$

In other words, we seek to find a modification r of image x that will minimize the distance of the modified image to the target class c_t.

Dodging In contrast to an attacker who wishes to impersonate a particular target, an attacker who wishes to dodge

[1] http://www.robots.ox.ac.uk/~vgg/software/vgg_face/

recognition is only interested in not being recognized as herself. To accomplish this goal, the attacker needs to find how to perturb the input x to minimize the probability of the class c_x. Such a perturbation r would maximize the value of $softmaxloss(f(x+r), c_x)$. To this end, we define the optimization problem for dodging as:

$$\underset{r}{\text{argmin}} \big(- softmaxloss(f(x+r), c_x) \big)$$

To solve these optimizations, we use the Gradient Descent (GD) algorithm [4]. GD is guaranteed to find a global minimum only when the objective is convex, but in practice often finds useful solutions regardless. In a nutshell, GD is an iterative algorithm that takes an initial solution to r and iteratively refines it to minimize the objective. GD iteratively evaluates the gradient, g, and updates the solution by $r = r - \alpha g$, for a positive value α (i.e., r is updated by "stepping" in the direction of the steepest descent). GD stops after convergence (i.e., changes in the objective function become negligible) or after a fixed number of iterations. In our work, we fix the number of iterations.

4.3 Facilitating Physical Realizability

Accomplishing impersonation or dodging in a digital environment does not guarantee that the attack will be physically realizable, as our experiments in Sec. 5.1 show. Therefore, we take steps to enable physical realizability. The first step involves implementing the attacks purely with facial accessories (specifically, eyeglass frames), which are physically realizable via 3d- or even 2d-printing technologies. The second step involves tweaking the mathematical formulation of the attacker's objective to focus on adversarial perturbations that are a) robust to small changes in viewing condition; b) smooth (as expected from natural images); and c) realizable by affordable printing technologies. In what follows, we describe the details of each step.

4.3.1 Utilizing Facial Accessories

While an attacker who perturbs arbitrary pixels that overlay her face can accomplish impersonation and dodging attacks, the perturbations may be impossible to implement successfully in practice (see Sec. 5.1). To address this, we utilize perturbed facial accessories (in particular, eyeglass frames) to implement the attacks. One advantage of facial accessories is that they can be easily implemented. In particular, we use a commodity inkjet printer (Epson XP-830) to print the front plane of the eyeglass frames on glossy paper, which we then affix to actual eyeglass frames when physically realizing attacks. Moreover, facial accessories, such as eyeglasses, help make attacks plausibly deniable, as it is natural for people to wear them.

Unless otherwise mentioned, in our experiments we use the eyeglass frames depicted in Fig. 1. These have a similar design to frames called "geek" frames in the eyewear industry.[2] We select them because of the easy availability of a digital model. After alignment, the frames occupy about 6.5% of the pixels of the 224×224 face images, similarly to real frames we have tested. This implies that the attacks perturb at most 6.5% of the pixels in the image.

To find the color of the frames necessary to achieve impersonation or dodging we first initialize their color to a solid color (e.g., yellow). Subsequently, we render the frames onto the image of the subject attempting the attack and iteratively update their color through the GD process. In each iteration, we randomize the position of the frames around a reference position by moving them by up to three pixels horizontally or vertically, and by rotating them up to four degrees. The rationale behind doing so is to craft adversarial perturbations that are tolerant to slight movements that are natural when physically wearing the frames.

[2]E.g., see: http://goo.gl/Nsd20I

Figure 1: The eyeglass frames used to fool FRSs (before texture perturbation). By Clker-Free-Vector-Images; source: https://goo.gl/3RHKZA.

4.3.2 Enhancing Perturbations' Robustness

Due to varying imaging conditions, such as changes in expression and pose, two images of the same face are unlikely to be exactly the same. As a result, to successfully realize the attacks, attackers need to find perturbations that are independent of the exact imaging conditions. In other words, an attacker would need to find perturbations that generalize beyond a single image, and can be used to deceive the FRSs into misclassifying many images of the attacker's face.

Thus far, the techniques to find perturbations were specific to a given input. To enhance the generality of the perturbations, we look for perturbations that can cause any image in a *set of inputs* to be misclassified. To this end, an attacker collects a set of images, X, and finds a single perturbation that optimizes her objective for every image $x \in X$. For impersonation, we formalize this as the following optimization problem (dodging is analogous):

$$\underset{r}{\text{argmin}} \sum_{x \in X} softmaxloss(f(x+r), l)$$

4.3.3 Enhancing Perturbations' Smoothness

Natural images (i.e., those captured in reality) are known to comprise smooth and consistent patches, where colors change only gradually within patches [24]. Therefore, to enhance plausible deniability, it is desirable to find perturbations that are smooth and consistent. In addition, due to sampling noise, extreme differences between adjacent pixels in the perturbation are unlikely to be accurately captured by cameras. Consequently, perturbations that are non-smooth may not be physically realizable.

To maintain the smoothness of perturbations, we update the optimization to account for minimizing *total variation* (TV) [24]. For a perturbation r, $TV(r)$ is defined as:

$$TV(r) = \sum_{i,j} \left((r_{i,j} - r_{i+1,j})^2 + (r_{i,j} - r_{i,j+1})^2 \right)^{\frac{1}{2}}$$

where $r_{i,j}$ are is a pixel in r at coordinates (i, j). $TV(r)$ is low when the values of adjacent pixels are close to each other (i.e., the perturbation is smooth), and high otherwise. Hence, by minimizing $TV(r)$ we improve the smoothness of the perturbed image and improve physical realizability.

4.3.4 Enhancing Perturbations' Printability

The range of colors that devices such as printers and screens can reproduce (the color *gamut*) is only a subset

of the $[0,1]^3$ RGB color space. Thus, to be able to successfully use a printer to realize adversarial perturbations, it is desirable to craft perturbations that are comprised mostly of colors reproducible by the printer. To find such perturbations, we define the *non-printability score* (*NPS*) of images to be high for images that contain unreproducible colors, and low otherwise. We then include minimizing the non-printability score as part of our optimization.

Let $P \subset [0,1]^3$ be the set of printable RGB triplets. We define the *NPS* of a pixel $\hat{p}$ as:

$$NPS(\hat{p}) = \prod_{p \in P} |\hat{p} - p|$$

If $\hat{p}$ belongs to P, or if it is close to some $p \in P$, then $NPS(p)$ will be low. Otherwise, $NPS(p)$ will be high. We intuitively generalize the definition of *NPS* of a perturbation as the sum of *NPS*s of all the pixels in the perturbation.

In practice, to approximate the color gamut of a printer, we print a color palette that comprises a fifth of the RGB color space (with uniform spacing). Subsequently, we capture an image of the palette with a camera to acquire a set of RGB triplets that can be printed. As the number of unique triplets is large ($\geq 10K$), the computation of *NPS* becomes computationally expensive. To this end, we quantize the set of colors to a set of 30 RGB triplets that have a minimal variance in distances from the complete set, and use only those in the definition of *NPS*; this is an optimization that we have found effective in practice.

In addition to having limited gamut, printers do not reproduce colors faithfully, i.e., the RGB values of a pixel requested to be printed do not correspond exactly to the RGB values of the printed pixel. To allow us to realize perturbations with high fidelity, we create a map m that maps colors (i.e., RGB triplets) requested to be printed to colors that are actually printed. We then utilize this map to realize perturbations with high fidelity. In particular, to print a perturbation as faithfully as possible, we replace the value p of each pixel in the perturbation with $\tilde{p}$, such that the printing error, $|p - m(\tilde{p})|$, is minimized. This process can be thought of as manual color management [20].

5. EVALUATION

We separately evaluate attacks that take place purely in the digital domain (Sec. 5.1) and physically realized attacks (Sec. 5.2).

5.1 Digital-Environment Experiments

We first discuss experiments that assess the difficulty of deceiving FRSs in a setting where the attacker can manipulate the digital input to the system, i.e., modify the images to be classified on a per-pixel level. Intuitively, an attacker who cannot fool FRSs successfully in such a setting will also struggle to do so physically in practice.

Experiment Description.

We ran eight dodging and impersonation attacks (see Sec. 4.2) on the white-box DNNs presented in Sec. 4.1. Between experiments, we varied (1) the attacker's goal (dodging or impersonation), (2) the area the attacker can perturb (the whole face or just eyeglass frames that the subject wears), and (3) the DNN that is attacked (DNN_A, DNN_B, or DNN_C). The frames we use are depicted in Fig. 1.

Figure 2: A dodging attack by perturbing an entire face. Left: an original image of actress Julia Jones (by Wenner Media; source: http://goo.gl/OFOIGB). Middle: A perturbed image for dodging. Right: The applied perturbation, after multiplying the absolute value of pixels' channels $\times 20$.

We measured the attacker's success (*success rate*) as the fraction of attempts in which she was able to achieve her goal. For dodging, the goal is merely to be misclassified, i.e., the DNN, when computing the probability of the image belonging to each target class, should find that the most probable class is one that does not identify the attacker. For impersonation, the goal is to be classified as a specific target, i.e., the DNN should compute that the most probable class for the input image is the class that identifies the target. In impersonation attempts, we picked the targets randomly from the set of people the DNN recognizes.

To simulate attacks on DNN_A and DNN_C, we chose at random 20 subjects for each experiment from the 2622 and 143 subjects, respectively, that the DNNs were trained on. We simulated attacks on DNN_B with all ten subjects it was trained to recognize. To compute statistics that generalize beyond individual images, we performed each attack (e.g., each attempt for subject X to impersonate target Y) on three images of the subject and report the mean success rate across those images. We ran the gradient descent process for at most 300 iterations, as going beyond that limit has diminishing returns in practice. A summary of the experiments and their results is presented in Table 1.

Experiment Results.

In experiments 1 and 2 we simulated dodging and impersonation attacks in which the attacker is allowed to perturb any pixel on her face. The attacker successfully achieved her goal in all attempts. Moreover, the adversarial examples found under these settings are likely imperceptible to humans (similar to adversarial examples found by Szegedy et al. [39]): the mean perturbation of a pixel that overlaid the face was 1.7, with standard deviation 0.93. An example is shown in Fig. 2.

We believe that the types of attacks examined in experiments 1 and 2 are far from practical. Because the perturbations are too subtle and lack structure, they may be too complicated to physically realize, e.g., it is likely to be impossible to modify a human face in exactly the way required by the attack. Moreover, the amount by which pixels are perturbed is often much smaller than the error that occurs when printing a perturbation and then capturing it with a scanner or a camera. Our experiments show that the average error per pixel in a perturbation when printed and scanned is 22.9, and its standard deviation is 38.22. Therefore, even if the attacker can successfully realize the perturbation, she is still unlikely to be able to deceive the system.

To this end, in experiments 3–8 we simulated dodging and impersonation attacks in which we perturbed only eyeglass frames of eyeglasses worn by each subject. With the

Experiment #	*Area perturbed*	*Goal*	*Model*	*# Attackers*	*Success rate*
1	Entire face	Dodging	DNN_A	20	100.00%
2	Entire face	Impersonation	DNN_A	20	100.00%
3	Eyeglass frames	Dodging	DNN_A	20	100.00%
4	Eyeglass frames	Dodging	DNN_B	10	100.00%
5	Eyeglass frames	Dodging	DNN_C	20	100.00%
6	Eyeglass frames	Impersonation	DNN_A	20	91.67%
7	Eyeglass frames	Impersonation	DNN_B	10	100.00%
8	Eyeglass frames	Impersonation	DNN_C	20	100.00%

Table 1: A summary of the digital-environment experiments attacking DNN_A, DNN_B, and DNN_C under the white-box scenario. In each attack we used three images of the subject that we sought to misclassify; the reported success rate is the mean success rate across those images.

Figure 3: An impersonation using frames. Left: Actress Kaylee Defer (by Fanpop; source: http://goo.gl/nTqqtF). Image classified correctly with probability 1. Middle: Perturbing frames to impersonate (actress) Nancy Travis. Right: The target (by Mingle Media TV; source: https://goo.gl/2BEx9a).

exception of experiment 6, the attacker was able to dodge recognition or impersonate targets in all attempts. In experiment 6 impersonators succeeded in about 91.67% of their attempts to fool DNN_A using perturbed eyeglasses. We hypothesize that due to the large number of classes DNN_A recognizes, the image space between some attacker-target pairs is occupied by other classes. Therefore, impersonating these targets requires evading several classification boundaries, making impersonation attacks more difficult.

Fig. 3 shows an example of a successful impersonation attempt using eyeglass frames.

5.2 Physical-Realizability Experiments

As discussed in Sec. 4.3, to physically realize an attack, we utilize a set of perturbed eyeglass frames, ensure that the perturbation is smooth and effective for misclassifying more than one image, and enhance the reproducibility of the perturbation's colors by the printing device. To achieve these goals and impersonate a target t, an attacker finds a perturbation by solving the following optimization problem:

$$\underset{r}{\operatorname{argmin}} \Big(\big(\sum_{x \in X} \mathit{softmaxloss}(x + r, c_t) \big) + \kappa_1 \cdot TV(r) + \kappa_2 \cdot NPS(r) \Big)$$

where κ_1 and κ_2 are constants for balancing the objectives and X is a set of images of the attacker. The formulation for dodging is analogous.

In this section we report on experiments for evaluating the efficacy of this approach in fooling DNN_B and DNN_C under semi-controlled imaging conditions.

Experiment Description.

The first three authors, whom DNN_B and DNN_C were trained to recognize, participated in the experiments;[3] we refer to them as subjects S_C, S_B, and S_A, respectively. For each subject we attempted two dodging attacks and two impersonation attacks—one of each type of attack on each of DNN_B and DNN_C. The targets in impersonation attacks were randomly selected (see Table 2).

We collected images of the subjects using a Canon T4i camera. To prevent extreme lighting variations, we collected images in a room without exterior windows. Subjects stood a fixed distance from the camera and were told to maintain a neutral expression and to make slight pose changes throughout the collection. While these collection conditions are only a subset of what would be encountered in practice, we believe they are realistic for some scenarios where FRS technology is used, e.g., within a building for access control.

For each subject, we collected 30–50 images in each of five sessions. In the first session, we collected a set of images that was used for generating the attacks (referred to as the set X in the mathematical representation). In this session, the subjects did not wear the eyeglass frames. In the second and third sessions, the subjects wore eyeglass frames to attempt dodging against DNN_B and DNN_C, respectively. In the fourth and the fifth sessions, the subjects wore frames to attempt impersonation against DNN_B and DNN_C.

We physically realized attacks by printing the eyeglass frames with an Epson XP-830 printer on Epson Glossy photo paper. Realizing these attacks is cheap and affordable; the approximate cost for printing an eyeglass frame is $0.22. Once printed, we cut out the frames and affixed them to the frames of an actual pair of eyeglasses. Examples of realizations are shown in Fig. 4.

To find the perturbation, the parameters κ_1 and κ_2 in the optimization were set to 0.15 and 0.25, respectively. The computational overhead of mounting attacks prohibited a broad exploration of the parameter space, but we found these values effective in practice. In addition, we limited the number of iterations of the GD process to 300.

Experiment Results.

To evaluate DNN_B and DNN_C in a non-adversarial setting, we classified the non-adversarial images collected in the first session. All the face images of the three subjects were

[3]The fourth author was excluded from the experiments due to logistical challenges posed by physical distance.

	Subject (attacker) info		Dodging results		Impersonation results			
DNN	*Subject*	*Identity*	*SR*	$E(p(\text{correct-class}))$	*Target*	*SR*	*SRT*	$E(p(\text{target}))$
DNN_B	S_A	3rd author	100.00%	0.01	Milla Jovovich	87.87%	48.48%	0.78
	S_B	2nd author	97.22%	0.03	S_C	88.00%	75.00%	0.75
	S_C	1st author	80.00%	0.35	Clive Owen	16.13%	0.00%	0.33
DNN_C	S_A	3rd author	100.00%	0.03	John Malkovich	100.00%	100.00%	0.99
	S_B	2nd author	100.00%	<0.01	Colin Powell	16.22%	0.00%	0.08
	S_C	1st author	100.00%	<0.01	Carson Daly	100.00%	100.00%	0.90

Table 2: A summary of the physical realizability experiments. To the left, we report the DNN attacked and the identity of the subjects (the attackers in the simulation). When not attempting to fool both DNNs, the subjects were originally classified correctly with mean probability >0.85. SR is the success rate. SRT is the success rate when using a threshold. $E(p(\text{class}))$ is the mean (expected) probability of the class when classifying all images. Results for S_C when attacking DNN_B were achieved with glasses that occupy 10% of the area of the image being classified; results for the other experiments were achieved with glasses that occupy 6% of the image.

classified correctly. The mean probability of the correct class across the classification attempts was above 0.85, which implies that naive attempts at impersonation or dodging are highly unlikely to succeed. In contrast, our experiments showed that an attacker who intentionally attempts to deceive the system will usually succeed. Table 2 summarizes these results. Fig. 4 presents examples of successful dodging attacks and impersonations.

Dodging All three subjects successfully dodged face recognition by wearing perturbed eyeglass frames. When wearing frames for dodging, as shown in Fig. 4a, all of S_A's images collected in the second and the third sessions were misclassified by DNN_B and DNN_C, respectively. When dodging, the mean probability DNN_B assigned to the S_A's class dropped remarkably from 1 to 0.01. Similarly, the mean probability assigned to c_{S_A} by DNN_C dropped from 0.85 to 0.03. Thus, the dodging attempts made it highly unlikely that the DNNs would output c_{S_A}, the correct classification result. S_B was able to dodge recognition in 97.22% of the attempts against DNN_B and in 100% of the attempts against DNN_C. When classifying S_B's images, the mean probability assigned by the DNNs to c_{S_B} became ≤ 0.03 as a result of the attack, also making it unlikely to be the class assigned by the DNN.

S_C's attempts in dodging recognition against DNN_C were also successful: all his images collected in the third session were misclassified, and the probability assigned to c_{S_C} was low (<0.01). However, when wearing a perturbed version of the frames shown in Fig. 1; none of S_C's images collected as part of the second session misled DNN_B. We conjecture that because S_C was the only subject who wore eyeglasses among the subjects used for training DNN_B, c_{S_C} became a likely class when classifying images showing people wearing eyeglasses. Therefore, it became particularly hard for S_C to fool DNN_B by perturbing only a small area of the image. Nevertheless, by increasing the size of the frames such that they occupied 10% of the area of the aligned image (frames shown in Fig. 5) it became possible for S_C to achieve physically realizable dodging at the cost of decreased inconspicuousness. Using the larger frames, S_C was able to dodge recognition in 80% of the images (mean probability of c_{S_C} dropped to 0.35).

Impersonation To simulate impersonation attempts, each subject was assigned two random targets: one for fooling DNN_B and one for fooling DNN_C. S_A, a 41-year-old white male, was assigned to impersonate Milla Jovovich, a 40-year-old white female, and John Malkovich, a 62-year-old white male; S_B, a 24-year-old South Asian female, was assigned to impersonate S_C, a 24-year-old Middle Eastern male, and Colin Powell, a 79-year-old white male; and S_C was assigned to impersonate Clive Owen, a 51-year-old male, and Carson Daly, a 43-year-old male.

Both of S_A's impersonation attempts were successful: 87.87% of his images collected in the fourth session were misclassified by DNN_B as Milla Jovovich (the mean probability of the target was 0.78), and all the images collected in the fifth session were misclassified as John Malkovich (mean probability of 0.99). In contrast, S_B and S_C had mixed success. On the one hand, S_B misled DNN_B by successfully impersonating S_C in 88% of her attempts (the mean probability of the target was 0.75), and S_C misled DNN_C into misclassifying him as Carson Daly in all of his attempts (mean probability of 0.99). On the other hand, they were able to successfully impersonate Colin Powell and Clive Owen[4] (respectively) only in about one of every six attempts. This success rate may be sufficient against, e.g., access-control systems that do not limit the number of recognition attempts, but may not be sufficient against systems that limit the number of attempts or in surveillance applications. We hypothesize that some targets are particularly difficult for some subjects to impersonate. We also believe, however, that even in those cases further refinement of the attacks can lead to greater success than we have so far measured.

In practice, to tune the security of the FRS, system operators may set a minimum threshold that the maximum probability in the DNN's output will have to exceed for a classification result to be accepted. Such a threshold balances security and usability: If the threshold is high, misclassification occurs less often, but correct classifications may be rejected, thus harming usability. On the other hand, for low thresholds, correct classifications will be accepted most of the time, but misclassifications are more likely to occur, which would harm security. Therefore, system operators usually try to find a threshold that balances the usability and the security of the deployed FRS. Using the test data, we found that by setting the threshold to 0.85, the false acceptance rate of DNN_B became 0, while true acceptance became 92.31%. This threshold strikes a good balance between usability and security [18]; by using it, false acceptance (of zero-effort im-

[4]Similarly to dodging, S_C required larger frames to impersonate the target.

(a) (b) (c) (d)

Figure 4: Examples of successful impersonation and dodging attacks. Fig. (a) shows S_A (top) and S_B (bottom) dodging against DNN_B. Fig. (b)–(d) show impersonations. Impersonators carrying out the attack are shown in the top row and corresponding impersonation targets in the bottom row. Fig. (b) shows S_A impersonating Milla Jovovich (by Georges Biard; source: https://goo.gl/GlsWlC); (c) S_B impersonating S_C; and (d) S_C impersonating Carson Daly (by Anthony Quintano; source: https://goo.gl/VfnDct).

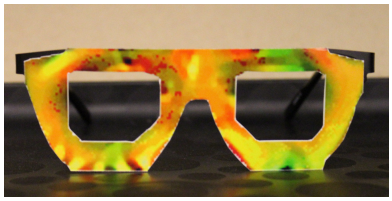

Figure 5: The eyeglass frames used by S_C for dodging recognition against DNN_B.

postors) never occurs, while true acceptance remains high. Following a similar procedure, we found that a threshold of 0.90 achieved a reasonable tradeoff between security and usability for DNN_C; the true acceptance rate became 92.01% and the false acceptance rate became 4e−3. Attempting to decrease the false acceptance rate to 0 reduced the true acceptance rate to 41.42%, making the FRS unusable.

Using thresholds changes the definition of successful impersonation: to successfully impersonate the target t, the probability assigned to c_t must exceed the threshold. Evaluating the previous impersonation attempts under this definition, we found that success rates generally decreased but remained high enough for the impersonations to be considered a real threat (see Table 2). For example, S_B's success rate when attempting to fool DNN_B and impersonate S_C decreased from 88.00% without threshold to 75.00% when using a threshold.

Time Complexity The DNNs we use in this work are large, e.g., the number of connections in DNN_B, the smallest DNN, is about 3.86e8. Thus, the main overhead when solving the optimization problem via GD is computing the derivatives of the DNNs with respect to the input images. For N_I images used in the optimizations and N_C connections in the DNN, the time complexity of each GD iteration is $\mathcal{O}(N_I * N_C)$. In practice, when using about 30 images, one iteration of GD on a MacBook Pro (equipped with 16GB of memory and a 2.2GHz Intel i7 CPU) takes about 52.72 seconds. Hence, running the optimization up to 300 iterations may take about 4.39 hours.

6. EXTENSION TO BLACK-BOX MODELS

So far we have examined attacks where the adversary has access to the model she is trying to deceive. In general, previous work on fooling ML systems has assumed knowledge of the architecture of the system (see Sec. 2). In this section we demonstrate how similar attacks can be applied in a black-box scenario. In such a scenario, the adversary would typically have access only to an oracle O which outputs a result for a given input and allows a limited number of queries. The threat model we consider here is one in which the adversary has access only to the oracle.

We next briefly describe a commercial FRS that we use in our experiments (Sec. 6.1), and then describe and evaluate preliminary attempts to carry out impersonation attacks in a black-box setting (Sec. 6.2–6.3).

6.1 Face++: A Commercial FRS

Face++ is a cross-platform commercial state-of-the-art FRS that is widely used by applications for facial recognition, detection, tracking, and analysis [46]. It has been shown to achieve accuracy over 97.3% on LFW [8]. Face++ allows users to upload training images and labels and trains an FRS that can be queried by applications. Given an image, the output from Face++ is the top three most probable classes of the image along with their confidence scores. Face++ is marketed as "face recognition in the cloud." Users have no access to the internals of the training process and the model used, nor even to a precise explanation of the meaning of the confidence scores. Face++ is rate-limited to 50,000 free queries per month per user.

To train the Face++ model, we used the same training data used for DNN_B in Sec. 4.1 to create a 10-class FRS.

6.2 Impersonation Attacks on Face++

The goal of our black-box attack is for an adversary to alter an image to which she has access so that it is misclassified. We attempted dodging attacks with randomly colored glasses and found that it worked immediately for several images. Therefore, in this section we focus on the problem of impersonation from a given *source* to a *target*.

We treat Face++ as an example black-box FRS, with its query function modeled as the oracle $O(x)$. The oracle returns *candidates*, an ordered list of three classes numbered from 1 to 3 in decreasing order of confidence.

Our algorithm for attacking Face++ uses particle swarm optimization (PSO) [6] as a subroutine. We begin by summarizing this technique.

Particle Swarm Optimization (PSO) Particle swarm optimization is a heuristic and stochastic algorithm for finding solutions to optimization problems by mimicking the behavior of a swarm of birds [6]. It iterates on a set of candidate solutions, called *particles*, and collectively called a *seed*, that it updates based on the evaluation of an objective function. Each particle is a candidate solution and occupies a *position* in the solution space. The *value* of a particle is the result of evaluating the objective function on that particle's position in the solution space. In each iteration, each particle is updated by adding a *velocity* to its position. The velocity is a weighted and randomized linear combination of the distance between (1) the current position of the particle and its best position thus far (P_{best}) and (2) the current position of the particle and the best position taken by *any* particle thus far (G_{best}), where "best" indicates that the objective function evaluates to the smallest value. Once a termination criterion is met, G_{best} should hold the solution for a local minimum.

We choose this over other black-box optimization methods such as surrogate models [3]—which require that the adversary has the training data and enough computational resources to train an effective surrogate—and genetic algorithms [2]—which though similar to PSO are much less computationally efficient [15, 34].

Since Face++ only returns the top three classes, PSO can make progress if *target* is in the top three results reported for any particle in the first iteration. However, when *target* is not in the top three for any set of initial solutions, the algorithm does not get any feedback on appropriate directions. To address this, we implement an algorithm we call *recursive impersonation* (Alg. 1). The goal of the algorithm to is reach the final target by attempting multiple intermediate impersonations on varying targets in successive runs of PSO. This is done in the hope that attempting an intermediate impersonation will move the swarm away from the previous solution space and toward candidate solutions that may result in the *target* being returned in the top three classes. If the target does not initially appear in the candidate list that results from querying the oracle with the original image, we select the class with the second highest confidence to be the intermediate target. On subsequent PSO runs, the most commonly occurring class labels that have not yet been used as targets become the new intermediate targets.

In our implementation, we modify the PSO subroutine to globally keep track of all the particles used in the last iteration of PSO as well as all particles throughout all PSO iterations for which invoking the oracle resulted in *target* appearing in the candidate list. The invoking algorithm has access to these saved particles and uses them in order to select a new intermediate impersonation target or to provide a seed for the next impersonation attempt.

On each run, PSO aims to minimize an objective function defined by $f(x + r)$, where r is the perturbation applied to the image x. $f(\cdot)$ is computed based on the output from the oracle O. The value of this objective at every particle is then used to move the swarm in a new direction during the next iteration of PSO. We experimented with several definitions of $f(\cdot)$. In practice, we found the following to be the most effective:

$$f(x) = \begin{cases} rank \cdot \frac{score_{top}}{score_{target}} & \text{if } target \in candidates \\ maxObjective & \text{if } target \notin candidates \end{cases}$$

The function uses the input x to query O for the *candidates*. The variable $score_{top}$ denotes the confidence score of the top-ranked item in *candidates*. If the *target* is in *candidates* then $score_{target}$ is its confidence score and *rank* its rank in *candidates*. When *target* is successfully impersonated, $f(\cdot)$ receives its minimal value of 1. If *target* is not in the top three candidates, the function should evaluate to *maxObjective*, which we set to be a sufficiently large value to indicate a result far less optimal than P_{best} or G_{best}.

Algorithm 1: Recursive Impersonation

```
Initialize epoch = 0, numParticles, epochs_max and seed.
Set candidates = O(image_original).
if target ∈ candidates then target_current = target;
else target_current = 2nd most probable class of candidates;
while epoch ≤ epoch_max do
    Run PSO subroutine with target_current and seed.
    if any particle impersonated target during PSO then
        solution was found. exit.
    else if target ∈ candidates of any query during PSO then
        target_current = target. Clear seed.
        seed ⊇ particles that produced this candidate from the
          current PSO run.
    else
        if new candidate emerges from current PSO run then
            target_current = new candidate. Clear seed.
            seed ⊇ particles that produced this candidate from
              the current PSO run.
        else
            no solution was found. exit.
        end
    end
    epoch = epoch + 1
end
```

6.3 Results

Experiment Description To evaluate our methodology, we picked four $\{source, target\}$ pairs from among the subjects on which Face++ was trained. Two of the pairs were chosen at random. The other two pairs were chosen as challenging impersonations by making sure that the target was not one of the top three classes reported by Face++ for the source image. To make the attack realistic in terms of possibility of physical realization, we restricted the perturbations to a pair of glasses. We did not attempt to physically realize the attack, however.

We ran our experiments with 25 particles. The particles were initialized by randomly generating 25 distinct pairs of glasses with smooth color variations. We ran the algorithm for a maximum of 15 epochs or attempts, and set the iteration limit of PSO to 50 and *maxObjective* to 50. We also assigned weights in computing the velocity such that a higher weight was given to G_{best} than P_{best}.

Experiment Results The results from the experiments are shown in Table 3. All attempted impersonations suc-

ceeded. Noteworthy is the low number of queries needed for the attacks, which shows that rate-limiting access to services will not always stop an online attack.

7. EXTENSION TO FACE DETECTION

In this section, we show how to generalize the basic approach presented in Sec. 4.2 to achieve *invisibility* to facial biometric systems. In an invisibility attack, an adversary seeks to trick an ML system not into misclassifying one person as another, but into simply failing to detect the presence of a person.

We examine this category of attacks for two reasons. First, most ML systems for identifying faces have two phases: detecting the presence of a face and then identifying the detected face. The detection phase is typically less closely tied to training data than the recognition phase. Hence, techniques to circumvent detection have the potential to apply more broadly across multiple systems.

Second, avoiding detection corresponds naturally to one type of motivation—the desire to achieve privacy. In seeking to achieve privacy, a person may specifically want to avoid causing culpability to be placed on another person. Similarly, a mislabeling of a face might be more likely to arouse suspicion or alert authorities than would the failure to notice the presence of a face at all, as might occur at an airport security checkpoint where faces detected by FDSs are confirmed against face images of passengers expected to travel, or faces of people wanted by the authorities.

In this work, we show how to perform invisibility attacks while attempting to maintain plausible deniability through the use of facial accessories. We defer the examination of the physical realizability of these attacks to future work.

7.1 The Viola-Jones Face Detector

As mentioned in Sec. 2, the Viola-Jones (VJ) face detector was designed with efficiency and accuracy in mind. The key idea to achieve both goals is to use a cascade of classifiers that have an ascending order of complexity. Each classifier is trained to detect the majority of the positive instances (presence of a face) and reject a large number of the negative instances. To detect an object in an image, several sub-windows are taken from the image and are evaluated by the detector. To be detected as a positive example, the sub-window needs to be classified as a positive example by all the classifiers in the cascade. On the other hand, being rejected by one classifier in the cascade results in classifying a sub-window as a negative example. Sub-windows that are rejected by simple classifiers are not further evaluated by the more sophisticated classifiers.

A classifier in the cascade is composed of a combination of *weak classifiers*. A weak classifier i is a simple classifier that outputs one of two possible values, $\tilde{a_i}$ or $\hat{a_i}$, based on one feature value, $f_i(\cdot)$, and a threshold b_i. Given a classifier that is composed of C weak classifiers, its decision function is defined as:

$$Classify(x) = \left(\sum_{i=1}^{C} \left((\tilde{a_i} - \hat{a_i})(f_i(x) > b_i) + \hat{a_i} \right) \right) > T$$

where T is the passing threshold of the classifier, x is the sub-window, and $f_i(x) > b_i$ evaluates to 1 if true and 0 otherwise.

As explained above, the VJ detector rejects a sub-window in case one of its classifiers rejects it. Thus, to evade detection it is sufficient to fool one cascade stage. Since the trained VJ is an open source (i.e., white-box) classifier [19], to find a minimal perturbation that can be used for evasion, we could potentially adapt and utilize the solution proposed by Szegedy et al. [39]. However, to solve the optimization problem, we need the classification function to be differentiable—as previously explained in Section 3.2—which $Classify(x)$ is not. Therefore, we utilize the *sigmoid* function, *sig* (as is often done in ML [37]), and formulate the optimization problem as:

$$\underset{r}{\operatorname{argmin}} \left(\left(\sum_{i=1}^{C} \left((\tilde{a_i} - \hat{a_i}) \cdot sig(k \cdot (f_i(x+r) - b_i)) + \hat{a_i} \right) - T \right) + c|r| \right) \quad (1)$$

where k is a positive real number that can be tuned to control the precision of the approximation. With this approximation, we can perform gradient descent to solve the optimization problem.

Figure 6: An example of an invisibility attack. Left: original image of actor Kiefer Sutherland. Middle: Invisibility by perturbing pixels that overlay the face. Right: Invisibility with the use of accessories.

7.2 Experiment Results

By generating a perturbation to evade a specific stage of the detector via the above technique, we are able to learn how to tweak pixel intensities in specific regions to successfully evade the whole cascade. To test this approach, we randomly selected 20 frontal images from the PubFig [21] face dataset, and tested whether each could be permuted to evade detection by fooling the first classifier in the cascade. We generated perturbed images as follows: we limited the perturbation to the area of the face, set c to 0.015 (as we found this to yield smaller perturbations in practice), and then performed line search on k to find the minimal perturbation necessary to evade the classifier, using a Limited BFGS [29] solver to solve the optimization (Eqn. 1).

For 19 out of the 20 images it was possible to evade detection. For the images that achieved evasion, the mean perturbation—the aggregate change in the value of the R, G, and B channels—of a pixel that overlays the face was 16.06 (standard deviation 6.35), which is relatively high and noticeable. As Fig. 6 shows, in some cases even the minimal perturbation necessary to evade detection required making changes to faces that could draw increased attention.

In another version of the attack, in which we sought to increase both the success rate and plausible deniability, we first added facial accessories specifically selected for their colors and contrast to the image; we then perturbed the image as in the previous attack. The accessories we used were: eyeglasses, a blond wig, bright eye contacts, eye blacks, and a winter hat. With this approach, it was possible to evade detection for all 20 face images. In addition, the amount by which each pixel needed to be perturbed dropped remark-

Source	*Target*	*Success rate*	*Avg. # queries*
Clive Owen	S_A	100%	109
Drew Barrymore	S_B	100%	134
S_D	S_C	100%	25
S_D	Clive Owen	100%	59

Table 3: Results of four attempted impersonation attacks, each run three times. S_A–S_C are the same subjects from Sec. 5.2. S_D is a 33-year-old Asian female. Each attempt had a different (randomized) initial seed and velocities. Number of queries is the total number of queries made of the oracle in the PSO iterations.

ably, thus contributing to plausible deniability. The mean perturbation of a pixel to avoid detection was 3.01 (standard deviation 2.12). An illustration of evasion attacks on VJ that utilize accessories is shown in Fig. 6.

8. DISCUSSION AND LIMITATIONS

Impact of Attacks As our reliance on technology increases, we sometimes forget that it can fail. In some cases, failures may be devastating and risk lives [36]. Our work and previous work show that the introduction of ML to systems, while bringing benefits, increases the attack surface of these systems (e.g., [23, 35]). Therefore, we should advocate for the integration of *only* those ML algorithms that are robust against evasion.

In this work we show that FRSs are vulnerable to a new type of attack: inconspicuous and physically realizable attacks that lead to dodging or impersonation. Such attacks can be especially pernicious as they can resist cursory investigation (at the least) and can provide attackers with plausible deniability. While we demonstrate the efficacy of these attacks on fooling one kind of DNN architecture, previous work has shown that adversarial examples can generalize beyond one DNN [39, 13]. Therefore, we believe our results imply that DNNs used for purposes beyond what we study may also be at risk.

Possible Defenses In future work, we plan to explore defenses to ameliorate risks caused by these attacks. Defenses explored in previous work either were ineffective in preventing adversarial examples (although they slightly harmed their imperceptibility) [13, 39], or were tested only on simple tasks (hand-written digit recognition) or on DNNs that are not state of the art [32]. We propose to defend against attacks differently. Adversarial attacks on vision systems exploit the fact that systems are sensitive to small changes in images to which humans are not. To address this, we propose to develop algorithms that reason about images more similarly to humans. In particular, we believe that approaches that classify images based on their attributes rather than on the intensities of their pixels may be effective (e.g., [21]).

Limitations The variations in imaging conditions that we investigate in this work are narrower than can be encountered in practice. For instance, we controlled lighting by taking images in a room that does not have external windows. These conditions are applicable to some practical cases (e.g., an FRS deployed within a building). However, other practical scenarios are more challenging, and effective attacks may have to be tolerant to a larger range of imaging conditions. For example, an attacker may not be able to control the lighting or her distance from the camera when an FRS is deployed in the street for surveillance purposes. We plan to investigate such challenging scenarios in future work.

In addition, the notion of inconspicuousness is subjective, and the only way to measure it adequately would include performing human-subject studies. We believe that the attacks demonstrated in this paper strongly suggest that it is possible to generate attacks that will remain unnoticeable by humans or will arouse no suspicion. We defer to future work achieving and evaluating attacks that meet this high standard for inconspicuousness. Furthermore, we will explore new directions to improve the color consistency, contrast, and texture-shape of the attacks we create, which is likely to enhance inconspicuousness.

9. CONCLUSION

In this paper, we demonstrated techniques for generating accessories in the form of eyeglass frames that, when printed and worn, can effectively fool state-of-the-art face-recognition systems. Our research builds on recent research in fooling machine-learning classifiers by perturbing inputs in an adversarial way, but does so with attention to two novel goals: the perturbations must be *physically realizable* and *inconspicuous*. We showed that our eyeglass frames enabled subjects to both *dodge* recognition and to *impersonate* others. We believe that our demonstration of techniques to realize these goals through printed eyeglass frames is both novel and important, and should inform future deliberations on the extent to which ML can be trusted in adversarial settings. Finally, we extended our work in two additional directions, first, to so-called *black-box* FRSs that can be queried but for which the internals are not known, and, second, to defeat state-of-the-art face *detection* systems.

10. ACKNOWLEDGEMENTS

We would like to thank Ariel Rao and Aya Fukami for volunteering their images and Matthew Fredrikson for discussions about adversarial machine learning.

11. REFERENCES

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7), 1997.

[2] A. D. Bethke. *Genetic Algorithms As Function Optimizers*. PhD thesis, University of Michigan, 1980.

[3] A. J. Booker, J. Dennis Jr, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural optimization*, 17(1):1–13, 1999.

[4] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proc. COMPSTAT*, 2010.

[5] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden voice commands. In *Proc. USENIX Security*, 2016.

[6] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proc. MHS*, 1995.

[7] N. Erdogmus and S. Marcel. Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect. In *Proc. IEEE BTAS*, 2013.

[8] H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou. Learning deep face representation. *arXiv preprint arXiv:1403.2802*, 2014.

[9] A. Fawzi, O. Fawzi, and P. Frossard. Fundamental limits on adversarial robustness. In *Proc. ICML, Workshop on Deep Learning*, 2015.

[10] R. Feng and B. Prabhakaran. Facilitating fashion camouflage art. In *ACM MM*, 2013.

[11] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proc. ACM CCS*, 2015.

[12] J. Galbally, C. McCool, J. Fierrez, S. Marcel, and J. Ortega-Garcia. On the vulnerability of face verification systems to hill-climbing attacks. *Pattern Recognition*, 43(3):1027–1038, 2010.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[14] A. Harvey. CV Dazzle: Camouflage from face detection. Master's thesis, New York University, 2010. Available at: http://cvdazzle.com.

[15] R. Hassan, B. Cohanim, O. De Weck, and G. Venter. A comparison of particle swarm optimization and the genetic algorithm. In *Proc. MDO*, 2005.

[16] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

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

[18] L. Introna and H. Nissenbaum. Facial recognition technology: A survey of policy and implementation issues. 2010. https://goo.gl/eIrldb.

[19] Itseez. OpenCV: Open Source Computer Vision. http://opencv.org/.

[20] N. Koren. Color management and color science. http://www.normankoren.com/color_management.html.

[21] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *Proc. ICCV*, 2009.

[22] Y. Li, K. Xu, Q. Yan, Y. Li, and R. H. Deng. Understanding OSN-based facial disclosure against face authentication systems. In *Proc. AsiaCCS*, 2014.

[23] B. Liang, M. Su, W. You, W. Shi, and G. Yang. Cracking classifiers for evasion: A case study on the Google's phishing pages filter. In *Proc. WWW*, 2016.

[24] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proc. CVPR*, 2015.

[25] Megvii Inc. Face++. http://www.faceplusplus.com/.

[26] MobileSec. Mobilesec Android Authentication Framework. https://github.com/mobilesec/authentication-framework-module-face.

[27] NEC. Face recognition. http://www.nec.com/en/global/solutions/biometrics/technologies/face_recognition.html.

[28] NEURO Technology. SentiVeillance SDK. http://www.neurotechnology.com/sentiveillance.html.

[29] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[30] OpenALPR. OpenALPR - Automatic License Plate Recognition. http://www.openalpr.com/.

[31] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *Proc. IEEE Euro S&P*, 2015.

[32] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proc. IEEE S&P*, 2016.

[33] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. BMVC*, 2015.

[34] L. M. Rios and N. V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[35] N. Rndic and P. Laskov. Practical evasion of a learning-based classifier: A case study. In *Proc. IEEE S&P*, 2014.

[36] P. Robinette, W. Li, R. Allen, A. M. Howard, and A. R. Wagner. Overtrust of robots in emergency evacuation scenarios. In *Proc. HRI*, 2016.

[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[38] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. CVPR*, 2015.

[39] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proc. ICLR*, 2014.

[40] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[41] A. Vedaldi and K. Lenc. MatConvNet – Convolutional neural networks for MATLAB. In *Proc. ACM MM*, 2015.

[42] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, 2001.

[43] G. L. Wittel and S. F. Wu. On attacking statistical spam filters. In *Proc. CEAS*, 2004.

[44] T. Yamada, S. Gohshi, and I. Echizen. Privacy visor: Method based on light absorbing and reflecting properties for preventing face image detection. In *Proc. SMC*, 2013.

[45] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proc. NIPS*, 2014.

[46] E. Zhou, Z. Cao, and Q. Yin. Naive-deep face recognition: Touching the limit of LFW benchmark or not? *arXiv preprint arXiv:1501.04690*, 2015.