Computer Vision and Image Understanding 202 (2021) 103103

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Detection of Face Recognition Adversarial Attacks

Fabio Valerio Massoli *, Fabio Carrara, Giuseppe Amato, Fabrizio Falchi

ISTI-CNR, via G. Moruzzi 1, 56124 Pisa, Italy

ARTICLE INFO

Keywords:
Deep Learning
Face Recognition
Adversarial attacks
Adversarial detection
Adversarial biometrics

ABSTRACT

Deep Learning methods have become state-of-the-art for solving tasks such as Face Recognition (FR). Unfortunately, despite their success, it has been pointed out that these learning models are exposed to *adversarial* inputs – images to which an imperceptible amount of noise for humans is added to maliciously fool a neural network – thus limiting their adoption in sensitive real-world applications. While it is true that an enormous effort has been spent to train robust models against this type of threat, adversarial detection techniques have recently started to draw attention within the scientific community. The advantage of using a detection approach is that it does not require to re-train any model; thus, it can be added to any system. In this context, we present our work on adversarial detection in forensics mainly focused on detecting attacks against FR systems in which the learning model is typically used only as features extractor. Thus, training a more robust classifier might not be enough to counteract the adversarial threats.

In this frame, the contribution of our work is four-fold: (i) we test our proposed adversarial detection approach against classification attacks, i.e., adversarial samples crafted to fool an FR neural network acting as a classifier; (ii) using a k-Nearest Neighbor (k-NN) algorithm as a guide, we generate deep features attacks against an FR system based on a neural network acting as features extractor, followed by a similarity-based procedure which returns the query identity; (iii) we use the deep features attacks to fool an FR system on the 1:1 face verification task, and we show their superior effectiveness with respect to classification attacks in evading such type of system; (iv) we use the detectors trained on the classification attacks to detect the deep features attacks, thus showing that such approach is generalizable to different classes of offensives.

1. Introduction

Deep Learning (DL) quickly occupied a central role in recent AI-related technological breakthroughs covering multiple fields and applications: vision (e.g., image classification Krizhevsky et al., 2012, object detection Girshick, 2015), natural language processing (Deng and Liu, 2018) and the combination of them (e.g., multi-modal Carrara et al., 2018a, and sentiment analysis Ortis et al., 2019). Despite achieving state-of-the-art (SotA) performance in many scenarios, Deep Neural Network (DNN) models still suffer from deficiencies that strongly limit their adoption in sensitive applications. Among others, the vulnerability in adversarial settings still poses challenges: it is relatively easy for an attacker to manipulate the output of a model by tampering its input often in an imperceptible way. The existence of these perturbed inputs – known as *adversarial examples* (Biggio et al., 2013; Szegedy et al., 2013) – constitutes one of the major roadblocks in security-related applications such as DL-based biometrics systems for surveillance and access control that, despite performing brilliantly in natural settings (Sundararajan and Woodard, 2018), can be easily evaded by knowledgeable adversaries.

Face Recognition (FR) enabled by DNN is a case in point. Several successful applications of deep models to FR have been proposed in the literature (Cao et al., 2018; Amato et al., 2018; Liu et al., 2017). Indeed, this kind of technology enables AI surveillance programs in multiple countries (Feldstein, 2019) and has already found its way into consumers' products (Sundararajan and Woodard, 2018). However, researchers already showed how adversarial attacks could jeopardize this kind of system both in the digital (Dong et al., 2019; Song et al., 2018) and physical domain (Sharif et al., 2016; Kurakin et al., 2016a).

Defensive approaches for adversarial attacks can be roughly categorized in two methodologies, namely *rectification* and *adversarial input detection*. In rectification methods, the goal is to recover the intended output of the model by increasing the robustness of the system, e.g., by trying to remove adversarial perturbation from the input (Li and Li, 2017; Liao et al., 2018) or by increasing the robustness of the model itself (Kurakin et al., 2016b; Papernot et al., 2016). On the other hand, adversarial detection aims at detecting an occurred attack by analyzing the behavior of the model (without changing it) and signaling anomalous events (Gong et al., 2017; Grosse et al., 2017; Amirian et al., 2018; Metzen et al., 2017). Notwithstanding, many of the proposed

* Corresponding author.
E-mail address: fabio.massoli@isti.cnr.it (F.V. Massoli).

https://doi.org/10.1016/j.cviu.2020.103103
Received 15 November 2019; Received in revised form 3 September 2020; Accepted 4 September 2020
Available online 7 September 2020

adversarial detection methods fall prey to strong adversaries too (Carlini and Wagner, 2017a). Novel techniques exploiting the training data manifold to ground the predictions of a model (Papernot and McDaniel, 2018; Carrara et al., 2019) exhibit good trade-offs between detection performance and resilience to attacks (Sitawarin and Wagner, 2019).

Facial recognition systems do not usually implement recognition based on deep-learning classifiers but rather follow a similarity-based approach: deep models are used to extract features from visual facial data, and decisions rely on similarity measurements among them. Indeed, standard benchmarks for facial recognition, such as IJB-B (Whitelam et al., 2017) and IJB-C (Maze et al., 2018), define two evaluation protocols, namely 1:1 face verification and 1:N face identification, based on such similarity procedures.

Sticking to those protocols, we provide an analysis of adversarial attacks and further detection in facial recognition systems, relying on SotA DL models acting as features extractors. In particular, we report the following contributions. First, we test our recently proposed detection technique (Carrara et al., 2018b) against classification attacks to a SotA FR system. Second, we generate deep features attacks, also named deep representations attacks, using a k-Nearest Neighbor (k-NN) algorithm as guidance, and we study their properties. Third, we use deep features attacks to fool an FR system on the face verification task, showing their superior effectiveness to classification attacks in fooling similarity-based systems. Finally, we use the detectors trained on classification attacks to detect deep features attacks, thus showing that our approach is generalizable to diverse types of attacks.

The rest of the paper is organized as follows. In Section 2, we briefly review some related works. In Section 3, we describe the algorithms used to craft adversarial examples, while in Section 4, we describe the adversarial detection technique used in our study. In Section 5, we present the results from our experimental campaign, and finally, in Section 6, we report our conclusions.

2. Related work

2.1. Adversarial attacks

After the seminal work of Szegedy et al. (2013), in which the authors studied adversarial examples against DNN, in the last years, an exploding growth in studies of adversarial attacks and defenses has been witnessed. The existence of adversarial examples for DNN was confirmed by researchers who proposed multiple crafting algorithms to find them efficiently since the early works. Among the most relevant attacking algorithms available in the literature, there are the box-constrained L-BFGS (Szegedy et al., 2013), FGSM (Goodfellow et al., 2014) and its variants (Kurakin et al., 2016a; Dong et al., 2018), and CW (Carlini and Wagner, 2017b). We dedicate Section 3 to a more detailed review of these algorithms, as we adopt them in this work to generate adversarial examples.

2.2. Face recognition adversarial attacks

FR is among the most relevant topics in computer vision. This field had drawn the attention of the scientific community since the early 90s when (Turk and Pentland, 1991) proposed the Eigenfaces approach. DL models, especially leveraging on the properties of Deep Convolutional Neural Network (DCNN), started to dominate this field since 2012, reaching performances up to 99.80% (Wang and Deng, 2018), thus overcoming human performance on this task. Despite the effort in training very robust DL models, such systems still show some weaknesses. For example, it has been shown that state-of-the-art face classifiers experience a performance drop when tested against cross-resolution images (Massoli et al., 2019, 2020). Moreover, they are vulnerable to adversarial attacks considering both the black-box (Dong et al., 2019) and white-box (Song et al., 2018; Sharif et al., 2016) settings. Concerning the attacks to face recognition systems, Sharif et al. (2016) demonstrated the feasibility and effectiveness of physical attacks by dodging recognition and impersonating other identities using eyeglass frames with a malicious texture. Dong et al. (2019) successfully performed black-box attacks on face recognition models and demonstrated their effectiveness in a real-world deployed system. Recent attacks on FR systems either exploit generative models obtaining a more natural perturbation (Song et al., 2018) or find natural adversarial examples by modifying identity-independent attributes (Qiu et al., 2019; Kakizaki and Yoshida, 2019), such as hair color, makeup, or the presence of glasses.

2.3. Adversarial defenses

Obtaining a system that is robust to adversarial examples turned out to be a challenging and still open task. The robustness of a model can be increased via adversarial training (Goodfellow et al., 2014; Huang et al., 2015), or model distillation (Papernot et al., 2015). In general, techniques that try to smooth, change, or hide the gradient surface of the model seen by an attacker called gradient-masking defenses, can increase the attack effort needed to find an adversarial example, but the enhanced model is still vulnerable to stronger attacks. Regularization methods have also been proposed to train robust DL models. As an example, (Yan et al., 2018) proposed to integrate a perturbation-based regularizer into the classification objective, thus penalizing the norm of adversarial perturbations. Another strategical direction involves detecting adversarial examples, which is creating robust systems composed of a vulnerable model and a detection system that signals occurring attacks. Detection subsystems are often implemented as binary detectors that discern authentic and adversarial inputs. Gong et al. (2017) proposed to train an additional binary classifier that decides whether an input image is pristine or tampered. Feinman et al. (2017) exploited Bayesian uncertainty available in dropout models and density estimates to spot attacks, while Yang et al. (2019) introduced a detection framework based on thresholding a scale estimate of feature attribution scores. Grosse et al. (2017) adopted statistical tests in the pixel space to demonstrate the discernibility of adversarial images and proposed introducing the "adversarial" class in the original classifier is contextually trained with the model. Similarly, Metzen et al. (2017) proposed a detection subnetwork that relies on intermediate representations constructed by the model at inference time. However, many detection schemes have been proven to be bypassable (Carlini and Wagner, 2017a).

Novel detection methods rely on the training data manifold, usually in the spaces defined by the intermediate representations of the network, for grounding the model prediction and detect anomalies. Carrara et al. (2019) and Papernot and McDaniel (2018) showed that a k-NN scheme based on intermediate representations of the training set could be used to define a score that measures the confidence of the classification produced by a deep model: such a score can then be used to filter out adversarial examples but also authentic errors occurring. Instead, Sotgiu et al. (2020) trained several RBF-SVMs on top of specific DNN layers to classify each extracted deep representation. The class scores generated by each of these shallow classifiers were then combined by another SVM in charge of the final adversarial detection. (Lu et al., 2017) propose SafetyNet, a detection framework that also employed SVM classifiers. Specifically, it leveraged an RBF-SVM that looked at quantized codes, obtained from ReLUs' output, to discern natural images from adversarial examples. Our detection approach exploits the intermediate representations too. However, differently from the above methods, it relies on the representations' evolution throughout the entire network, also referred to as trajectories, represented by a sequence of distance embeddings (a detailed description of our method is reported in Section 4).

To our knowledge, the most relevant work that copes with detecting tampered facial recognition is Goswami et al. (2019), in which the authors attacked facial recognition systems in a classification setting

and devised a detection approach to decide whether to recover the original input. In the detection part, they proposed to compare intermediate network activations to their average values defined over a training set, and used layer-wise distances as features in a two-class SVM adversarial detector. However, their analysis only included the recognition-by-classification setting, while we covered additional real-world settings, such as attacks on k-NN identification and verification systems.

3. Adversarial attacks

In this section, we detail some of the most famous algorithms used to craft adversarial examples.

3.1. L-Broyden–Fletcher–Goldfarb–Shanno

Szegedy et al. (2013) formalized the adversarial attack as an optimization problem solved using the L-BFGS algorithm, expressed as:

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & c \cdot \| \boldsymbol{\delta} \|_2 + J_\theta(\mathbf{x}+\boldsymbol{\delta},\ t) \\ \text{subject to} \quad & L^m \leq \mathbf{x}+\boldsymbol{\delta} \leq U^m\,, \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the input image, J_θ is the objective function of the model with parameters θ, δ is the adversarial perturbation, t is the desired output label, $[L,\ U]^m$ represents the allowed pixel range, and $c(>0)$ is evaluated by performing line-search.

3.2. Fast gradient sign method

The Fast Gradient Sign Method (Goodfellow et al., 2014) (FGSM) is a one-step method in which the perturbation is found by following the direction of the gradient $\nabla_{\mathbf{x}} J_\theta$ of the objective function used to train the DL model with respect to the input image $\mathbf{x}$. The adversarial example is then given by:

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} J_\theta(\mathbf{x}, y))\,, \tag{2}$$

where y is the class label for $\mathbf{x}$ and ϵ is the maximum distortion allowed on the input such that $\| \mathbf{x} - \mathbf{x}_{adv} \|_\infty < \epsilon$.

3.3. Basic iterative method

The Basic Iterative Method (Kurakin et al., 2016a) (BIM) applies the FGSM (Goodfellow et al., 2014) attack multiple times with small step size. It is given by:

$$\mathbf{x}^{adv}_{N+1} = Clip_{\mathbf{x},\epsilon}\left\{\mathbf{x}^{adv}_N + \alpha \cdot \text{sign}(\nabla_{\mathbf{x}} J_\theta(\mathbf{x}^{adv}_N, y))\right\}, \tag{3}$$

where $\mathbf{x}^{adv}_0 = \mathbf{x}$, the $Clip(\cdot)$ function clips the values of the pixels to the allowed range, and α is the step size. An improved version of the attack, named Projected Gradient Descent (Madry et al., 2017) (PGD), starts the iteration procedure from an acceptable random perturbation of $\mathbf{x}$.

3.4. Momentum iterative-FGSM

The MI-FGSM method (Dong et al., 2018) is an iterative procedure based on the substitution of the current gradient with the accumulated ones from all the previous steps. The velocity vector in the gradient direction is given by:

$$\mathbf{g}_{N+1} = \mu \cdot \mathbf{g}_N + \frac{J_\theta(\mathbf{x}^{adv}_N, y)}{\| \nabla_{\mathbf{x}} J_\theta(\mathbf{x}^{adv}_N, y) \|_1}\,, \tag{4}$$

where $\mathbf{x}^{adv}_0 = \mathbf{x}$, $\mathbf{g}_0 = 0$, and μ is the decay factor of the running average. Thus, the adversarial in the ϵ-vicinity measured by L_2 distance is given by:

$$\mathbf{x}^{adv}_{N+1} = \mathbf{x}^{adv}_N + \alpha \cdot \frac{\mathbf{g}_{N+1}}{\| \mathbf{g}_{N+1} \|_2}\,, \tag{5}$$

where $\alpha = \epsilon/T$ with T being the total number of iterations.

3.5. Carlini-Wagner

Carlini and Wagner (2017b) (CW) proposed three gradient-based attacks each based on a different distance metric, namely L_0, L_2 and L_∞ attacks.

Given an input $\mathbf{x}$ and a target class t, the L_2 attack is given by:

$$\min \left\| \frac{1}{2}(\tanh(\mathbf{w})+1) - \mathbf{x} \right\|_2^2 + c \cdot f\left(\frac{1}{2}(\tanh(\mathbf{w})+1)\right)$$

with

$$f(\mathbf{x}^{adv}) = \max(\max\{\mathbf{Z}(\mathbf{x}^{adv})_i : i \neq t\} - \mathbf{Z}(\mathbf{x}^{adv})_t, -k)\,, \tag{6}$$

where f is the objective function, $\mathbf{Z}(\cdot)$ are the logits before the softmax layer, $\mathbf{w}$ is the variable to optimize over, and k is a parameter that allows controlling the confidence with which the misclassification occurs.

Concerning the L_∞ attack, it is not fully differentiable, and the standard gradient descent does not perform well for it. Eq. (7) shows the L_∞ version of the attack:

$$\min\ c \cdot f(\mathbf{x}+\boldsymbol{\delta}) + \sum_i [(\delta_i - \tau)^+]\,, \tag{7}$$

where τ is a threshold value for the adversarial perturbation. Finally, the L_0 attack is based on the idea of iteratively using L_2 to find a minimal set of pixels to be modified to generate an adversarial example.

3.6. Deep features attack

All the attack algorithms mentioned above focus their attention towards the classification output. Differently, Sabour et al. (2015) proposed an approach finalized to find an adversarial perturbation able to generate a deep representation as close as possible to the natural sample's one. The procedure starts by selecting a source and a guide images, $\mathbf{I}_s$ and $\mathbf{I}_g$, respectively. The goal of the adversary is to perturb $\mathbf{I}_s$ to generate a new image $\mathbf{I}_\alpha$ which internal representation at a specific layer k of the threatened model, $\phi_k(\mathbf{I}_\alpha)$, has a small Euclidean distance from $\phi_k(\mathbf{I}_g)$. At the same time, $\mathbf{I}_\alpha$ has to be close to $\mathbf{I}_s$ in the pixels space. Specifically, $\mathbf{I}_\alpha$ is defined to be the solution to the constrained optimization problem:

$$\begin{aligned} \mathbf{I}_\alpha &= \arg\min_{\mathbf{I}} \| \phi_k(\mathbf{I}) - \phi_k(\mathbf{I}_g) \|_2^2, \\ \text{subject to} \quad & \| \mathbf{I} - \mathbf{I}_s \|_\infty < \delta\,, \end{aligned} \tag{8}$$

where δ is the maximum allowed perturbation on the source image.

4. Adversarials detection approach

Our work aims at finding a procedure to empower a DL-based system to detect adversarial attacks. We posit our approach leveraging the piece-wise functional structure of DNN models. Specifically, a generic classifier $f_\theta(\cdot) : \mathcal{X} \to \mathcal{C}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ with d being the input dimensionality and $\mathcal{C}$ is the set of allowed labels, can be represented as a sequence of layers each applying a specific transformation to its input:

$$f_\theta(\mathbf{x}) = f^n(\mathbf{o}_{n-1}; \theta_n) \circ f^{n-1}(\mathbf{o}_{n-2}; \theta_{n-1}) \circ \ldots \circ f^0(\mathbf{x}; \theta_0)\,, \tag{9}$$

where f^i represents the ith layer with parameters θ_i, $\mathbf{o}_i$ is the output the ith layer, and $\mathbf{x}$ is the classifier input. Our detection approach exploits the idea that the evolution of the features maps might differ between manipulated and not-manipulated samples during the forward step of the threatened model. To capture the evolution of the features, we leverage the concept of distance space in which the coordinates represent the distances of the input from specific reference points, also called pivots. We embed the features extracted from the input at intermediate layers into several features distance spaces, one for each layer, obtaining a sequence that can be interpreted as describing the

input trajectory. Thus, by tracing the different behavior exhibited by natural images and manipulated samples, we can discern among them.

As the first step in our approach, we define the features distance spaces by evaluating the pivots. Given the set of layers $\mathcal{L} = \{l_i \mid i = 1, 2, \dots, L\}$ and the set of classes $C = \{c_j \mid j = 0, 1, \dots, C\}$, we define the pivots $\mathbf{p}_i^j$ as the "likely" position of a natural sample's features, belonging to class j, extracted at layer i. Once the pivots has been fixed, we can embed the deep representations of an input image. Given an output activation map at layer i, $\mathbf{o}_i$, its pivoted embedding is given by:

$$\mathbf{e}_i = \left(d\left(\mathbf{o}_i, \mathbf{p}_i^0\right),\ d\left(\mathbf{o}_i, \mathbf{p}_i^1\right),\ \dots,\ d\left(\mathbf{o}_i, \mathbf{p}_i^C\right) \right) \in \mathbb{R}^C, \tag{10}$$

where $\mathbf{p}_i^j$ represents the pivot relative to layer i and class j. The sequence of the embeddings of each layer, $\{\mathbf{e}_i \mid i = 1, 2, \dots, L\}$, can be compactly represented as the matrix $\mathbf{E} = \left\{ d\left(\mathbf{o}_i, \mathbf{p}_i^j\right) \right\}_{ij} \in \mathbb{R}^{|\mathcal{L}| \times |\mathcal{C}|}$ that represents the input to our adversarial detector. As the distance function $d(\cdot, \cdot)$, we explore the L_2 distance and the cosine similarity.[1]

Concerning the pivots, we evaluate them on the training set only, and we consider two different choices, namely class centroids and class medoids. We report the two formulations in Eq. (11):

$$\mathbf{p}_i^j = \mathbf{c}_i^j = \frac{1}{|B_c|} \cdot \sum_n^{|B_c|} \mathbf{o}_{i,n}^j,$$

or

$$\mathbf{p}_i^j = \mathbf{m}_i^j = \arg\min_{\mathbf{o} \in \mathcal{O}_i^j} \sum_n^{|B_c|} \| \mathbf{o} - \mathbf{o}_{i,n}^j \|_2^2, \tag{11}$$

where $\mathbf{c}_i^j$ and $\mathbf{m}_i^j$ are the centroid and medoid, respectively, for class j evaluated at the layer i, $|B_c|$ represents the class cardinality, and $\mathbf{o}_{i,n}^j$ is the output at layer i of class j of the nth training sample. Moreover, to reduce memory requirements, we applied an average pooling operation to each features map, thus reducing the size of each deep representation to $n_c \times 1 \times 1$, where n_c is the number of channels.

As a source dataset, we use the VGGFace2 (Cao et al., 2018) test set comprising 500 identities shared among ~170K images. To train and test the detector, we randomly extracted 20 images for each class while we use all the remaining ones to evaluate the class pivots. We split the 20 images into two halves so that we use 10 images as natural samples and 10 for adversarial generation only. Thus, we end up with 5000 malicious inputs for each adversarial configuration considered and with 5000 natural samples. More details on the adversarial generation are given in Section 5.

Our threatened model is the SotA Se-Net-50 from (Cao et al., 2018). Sixteen bottleneck layers characterize the model architecture (He et al., 2016), and we exploit their outputs to embed the evolution of the features. Precisely, considering that our dataset comprises 500 identities, each corresponding to a different class, we end up with an embedding $\mathbf{E} \in \mathbb{R}^{16 \times 500}$, for each image, that we used as input for our detector.

We examine two different architectures for the adversarial detector: a Multi-Layer Perceptron (MLP) and a Long-Short Term Memory (LSTM) network. For the MLP, we flatten the input matrix to obtain an input vector of size 8000, while for the LSTM, we consider the input as a 16-length sequence of 500-dimensional features. For both models, we set their hyperparameters (the number of inner layers and their size for the MLP and the hidden state size for the LSTM) through grid search. The optimal architecture for the MLP comprises one hidden layer with 100 units followed by a ReLU activation function and a Dropout, while the best LSTM is bidirectional and has a hidden state size equals to 100. Concerning both types of detectors, their output is fed into a Fully Connected (FC) layer, followed by a sigmoid activation function. Fig. 1 shows a schematic view of the entire system.

[1] When a similarity function is used to perform the embedding, we are building a feature similarity space instead of a distance space.

5. Experimental results

In this section, we report the experimental results we obtained so far. First, we focus on the detection of the attacks against a SotA FR model acting as a classifier. We craft adversarial examples employing known algorithms (Section 3), and then we train and test our detectors against them (Section 5.1). Afterward, in Section 5.2, we generate deep features (DF) attacks (Section 3.6) against an FR system based on the similarity among image descriptors extracted by a DCNN. To generate DF attacks, we use a k-NN algorithm as guidance in the optimization procedure simulating the similarity-based mechanism used to recognize people in real-world applications. We study these attacks, and in particular, we show that they represent a greater threat, compared to classification attacks, concerning FR systems. We then use DF adversarial examples to evade a face verification protocol (Section 5.3), and we compare their effectiveness with the CW (Carlini and Wagner, 2017b) attack. Subsequently, we use our detector, trained on classification attacks, to detect DF attacks, thus showing the generalization property of our detection approach. Then, we commit Section 5.4 to the defense-aware adversary setting, in which we study our detection system's resilience against white-box attacks.

Finally, in Section 5.5, we conduct a second experimental campaign in which we compare our approach against SotA detection algorithms on the MNIST (LeCun et al., 1998a,a) and CIFAR-10 (Krizhevsky et al., 2009a,b) datasets. Moreover, concerning the former dataset, we test our approach against a defense-aware adversary in this case too and compare our results with others available in the literature.

The code relative to our detection method and experiments is publicly available on *github*.[2]

5.1. Classification attacks detection

As we mentioned above, in this section we focus on classification attacks. The threatened model is the SotA facial recognition model from (Cao et al., 2018) trained on the 8631 classes of the VGGFace2 (Cao et al., 2018) train set. Instead, in our experiment, we use the 500 identities contained in its test set. Thus, we replace the classifier layer with a 500-ways FC layer, and we train it employing the SGD optimizer with a batch size of 256 and a learning rate of 10^{-3} halved every time the loss plateaus. As a preprocessing step, we resize the images so that the shortest side measured 256 pixels. Afterward, we randomly crop a 224×224 region of the image, and we subtracted the average pixel value channel-wise. For model evaluation, we use the same preprocessing except that a central crop substitutes the random one.

To craft adversarial examples, we use the *foolbox*[3] implementation of the MI-FGSM (Dong et al., 2018) (with L_∞ norm), the BIM (Kurakin et al., 2016b) and the CW (Carlini and Wagner, 2017b), both with L_2 norm, attacks. As far as the first two are concerned, we consider a maximum perturbation $\epsilon \in \{0.03, 0.07, 0.1, 0.3\}$ and a number of maximum iterations $\in \{30, 50\}$. Instead, for the CW attack, we adopt the implemented default value of the parameters, i.e., 5 binary search steps and 1000 iterations. As mentioned in Section 4, to train the detectors, we select 10K correctly classified images, 20 for each identity, and split them into two halves. We use one half as natural samples and the other half to craft adversarial inputs for each attack configuration.

To train the detectors, we run the Adam optimizer (Kingma and Ba, 2014) for 150 epochs, we set the batch size to 256 and the learning rate to 5×10^{-3}. To balance the sample distribution within mini-batches, we employ a weighted random sampler, thus avoiding bias towards attacks with higher cardinality. We train each architecture (Section 4) with all the four possible combinations of pivot types, centroids or medoids, and embedding metrics, L_2 distance or cosine similarity. We test the

[2] https://github.com/fvmassoli/trj-based-adversarials-detection.

[3] https://foolbox.readthedocs.io/en/stable/.

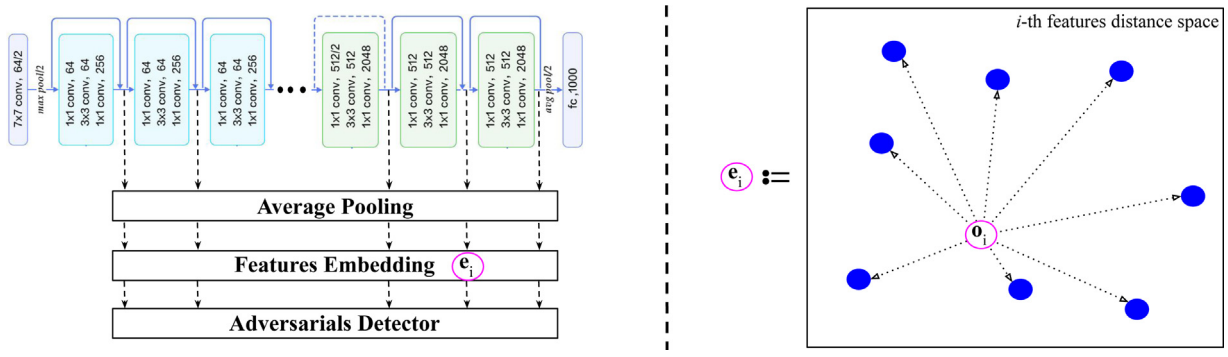

Fig. 1. Schematic view of the proposed detection algorithm. Left: model architecture and embedding procedure. The magenta circle represents a single embedding vector $\mathbf{e}_i$. Right: representation of the ith features distance space. Given a features map $\mathbf{o}_i$, we evaluate its embedding $\mathbf{e}_i$ as the vector whose components are the distances between the current features and the pivots (blue circles), centroids or medoids, of each class. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

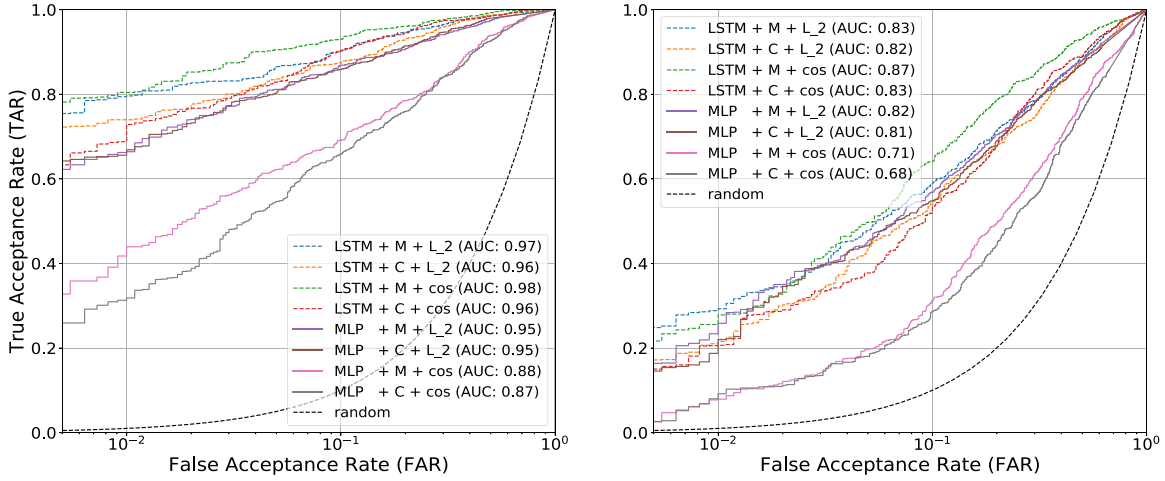

Fig. 2. ROCs for each architecture (LSTM or MLP), class representative (centroids or medoids), and distance metric (L_2 distance or cosine similarity) combination. Left: targeted attacks. Right: untargeted attacks.

Table 1
AUC values for each configuration reported in Fig. 2. The last column is a summary of the single-attacks AUCs. We emphasize in bold the performance of the best model.

Configuration	BIM		CW		MI-FGSM		Macro-AUC	
	Targ.	Untarg.	Targ.	Untarg.	Targ.	Untarg.	Targ.	Untarg.
LSTM + M + L_2	0.977	0.878	0.871	0.615	0.986	0.889	0.944	0.794
LSTM + C + L_2	0.970	0.863	0.857	0.596	0.982	0.869	0.936	0.776
LSTM + M + cos	**0.986**	**0.929**	**0.904**	**0.599**	**0.991**	**0.930**	**0.960**	**0.819**
LSTM + C + cos	0.968	0.884	0.895	0.568	0.981	0.886	0.948	0.779
MLP + M + L_2	0.964	0.885	0.793	0.559	0.979	0.882	0.912	0.775
MLP + C + L_2	0.962	0.874	0.808	0.557	0.979	0.874	0.916	0.768
MLP + M + cos	0.890	0.763	0.668	0.460	0.940	0.769	0.832	0.664
MLP + C + cos	0.868	0.730	0.720	0.467	0.915	0.739	0.834	0.645

detectors against targeted and untargeted attacks separately, and we report the relative Receiving Operating Characteristics (ROC) curves in Fig. 2 and, as a summary, the Area Under the Curve (AUC) in Table 1.

Table 1 shows promising results, thus proving the effectiveness of our approach. It is worth to notice that the untargeted attacks are, on average, more challenging to detect respect to the targeted ones. A possible explanation is that these attacks typically find the closest adversarial examples to the input image. Thus, an embedding method based on the distance between representation may have difficulties in detecting them. We further this intuition in Section 5.2.

5.2. Deep features attacks

We now move our focus onto the main subject of our study, i.e., attacks against real-world FR systems. First, we study the deep features attacks' properties showing their suitability to fool similarity-based systems. Differently from the classification setting, as in Section 5.1, in typical real-world applications, a DL model does not perform the recognition of faces by itself. Instead, an FR system exploits the ability of DNNs to generate discriminative face descriptors that are then compared, through similarity measurements, to fulfill the recognition task. Such a procedure is commonly adopted, for example, when testing

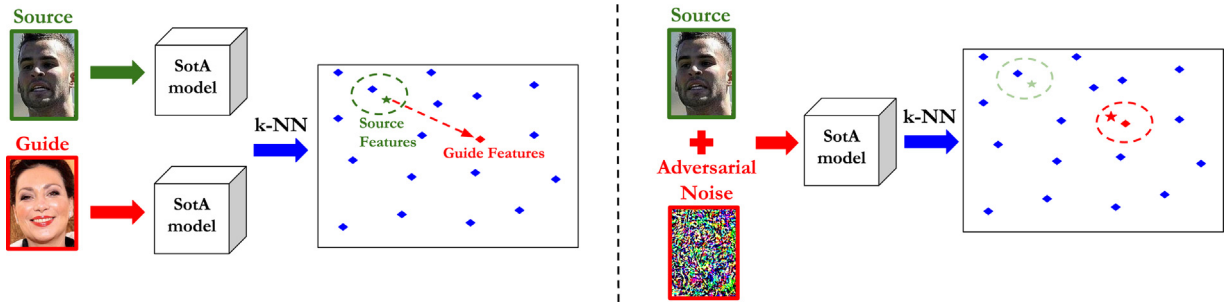

Fig. 3. Schematic view of the adversarial generation procedure considering a state-of-the-art (SotA) model as features extractor and a k-NN to assess the face identity. Left, before the attack. Right, after the attack.

FR model performance on the IJB-B (Whitelam et al., 2017) and IJB-C (Maze et al., 2018) benchmark datasets. Given a similarity-based setting, adversaries that focus only on the DL output might not succeed in fooling the whole system. Thus, we need a different approach to evade it.

As described in Section 3.6, one can use the distance among deep representations as a guiding principle to craft adversarial examples instead of the wrong label assignment. Thus, nurturing this idea, we synthesize malicious inputs by using the distance among deep features as a guidance (Sabour et al., 2015). To this end, we formulate the optimization procedure such that the face descriptors generated from the adversarial can fool the final similarity-based algorithm.

We start our experimental campaign by considering an FR system that relies on a DCNN to generate face descriptors, and we exploit a k-NN classifier to play the role of the similarity-based algorithm that accomplishes the final task of recognizing faces.

To our aim, we use the SotA model from (Cao et al., 2018) as a features extractor, and we evaluate the templates for the 500 classes in the VGGFace2 (Cao et al., 2018) test set to construct the identities database. Each identity in the database is represented by a template vector, i.e., a vector of features obtained by averaging several deep representations extracted from different images of the same person. The adversarial generation is formulated as an optimization problem (Sabour et al., 2015) solved using the L-BFGS-B algorithm. The constraint is used to threshold the maximum perturbation, δ, on the original image's pixels. To adapt the generation of the adversarial examples to our needs, we use a k-NN classifier as guidance through the optimization procedure. The optimization is then stopped once the targeted or untargeted attack's objective is met; that is, the k-NN had classified the adversarial as belonging to the guide-image class or merely misclassified the face image, considering targeted and untargeted attacks respectively. A schematic view of the algorithm is shown in Fig. 3.

In our experiments we consider a maximum allowed perturbation $\delta \in \{5.0, 7.0, 10.0\}$, and we report an example of adversarial samples in Fig. 4.

From Fig. 4, we notice that the generated images look equal to the original ones, i.e., there is not an evident trace of the guide image into the adversarial one. Considering targeted attacks, we obtained a success rate of **95.6%**, **96.2%** and **96.3%** considering a value for $\delta = 5.0, 7.0, 10.0$, respectively. Instead, concerning untargeted attacks we obtained **96.8%** success rate for $\delta = 5.0, 7.0, 10.0$. For all the attacks, we set the maximum number of iterations to 700.

Even if already mentioned above, we want to stress again that, in contrast to Section 5.1, we ground the current adversarial generation on similarities among deep representations. Indeed, since the FR systems exploit a DNN model to generate face descriptors only, it is not guaranteed that even if a model misclassifies a face image, the deep adversarial representation is close enough to the descriptor of the wrong class from a similarity-based perspective.

In Fig. 5 we report an interesting result to justify our intuition. The figure shows the distribution of the distance among the adversarial examples and the centroids of their relative classes considering classification attacks (BIM Kurakin et al., 2016a, MI-FGSM Dong et al., 2018, and CW Carlini and Wagner, 2017b) and DF attacks (Sabour et al., 2015), using a k-NN as guidance.

Fig. 4. Adversarial examples for two different values of the threshold applied while solving the optimization problem. Top: $\delta = 5.0$. Bottom: $\delta = 10.0$.

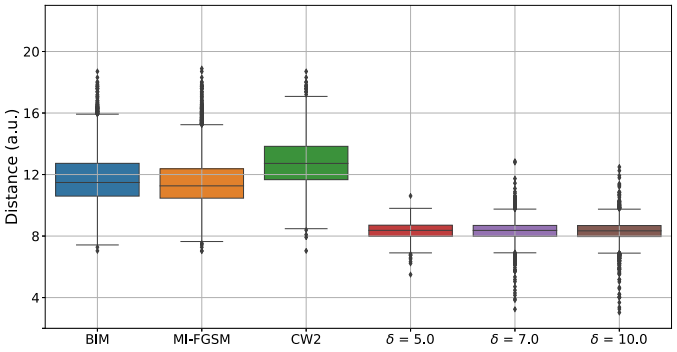

Fig. 5. Euclidean distance among deep features of adversarial examples and the assigned class centroid considering each targeted attack singularly. The "δ" values correspond to the maximum L_∞ perturbation allowed, for each pixel, for the k-NN-guided attacks.

As we can see from Fig. 5, even though classification attacks can fool a DCNN, the distance among the representations of adversarial classification samples and the centroid of the target classes is larger than the one obtained by deep representation-based attacks. Thus, the latter represents a more significant threat than the former types of attacks for a real-world FR system. The same behavior is observed for untargeted attacks.

Concerning Fig. 5, we can also notice that the average distance among the adversarial examples from the class centroid is quite stable among the three different values of the threshold δ. We can explain such behavior by observing that in most of the cases, we have $\delta \leq 5$ even though we set a larger threshold. Thus, a higher threshold had the effects that only a small portion of the image is perturbed.

Specifically, considering the values of $\delta \in \{5.0, 7.0, 10.0\}$, the percentage of pixels whose perturbation is within an L_∞ distance of 5.0 is 88.3%, 85.6%, 84.7% respectively. This behavior is shown in Fig. 6 for targeted attacks. Similar results hold in the untargeted setting.

5.3. Face verification

Having shown the effectiveness of DF attacks in fooling a real-world FR system, we now directly compare their evasion ability with the one of classification attacks. Finally, we test our detection approach against them even though we train the detectors against classification attacks only. Specifically, we consider an FR system tested against the 1:1 face verification protocol as the threatened model. In such a scenario, the goal of the system is to compare two face images to claim if they belong to the same identity or not.

In the DL context, such a decision is typically based upon similarity measurements among deep features extracted from the query images. Especially, given a trained model, its ROC is first evaluated to estimate the threshold to be used as a reference value for the similarity score to assess if two faces belong or not to the same identity. A real-world application example of this kind of protocol is a restricted access area control system. Since, in this type of application, the False Positives pose a more significant threat than the False Negatives, it is crucial to evaluate the ROC curve down to low values of the False Acceptance Rate. Such a demand translates into the requirement for assessing the similarity scores among a larger number of negative pairs to the positive ones.

To our aim, we adopt the SotA DCNN from (Cao et al., 2018), and we evaluate the cosine similarity among features vectors as similarity measurement. From our analyses, we obtain a ROC curve with an AUC value equals to 99.03%. We then used the Equal Error Rate (EER) threshold, equals to 0.448, as a reference value for the similarity measurement. Concerning the attacks, we hypothesize two scenarios:

- *Impersonation Attack*. The attacker tries to fool the system by leading it to falsely predict that two face images belong to the same identity. This situation emulates an intruder who tries to enter a restricted area or, in a more general case, when someone is made recognizable as a different person.
- *Evading Attack*. As opposed to the previous case, now the attacker seeks to fool the system leading it to predict that two face images, corresponding to the same identity, belong to the different people. This circumstance emulates the condition of someone whose identity is made unrecognizable.

In the former case, two images that belong to different people have to be "equal enough", i.e., their similarity measurement has to be above the evaluated threshold, while in the latter case, two images have to be "distant enough", i.e., below the threshold.

To craft adversarial examples, we consider the CW (Carlini and Wagner, 2017b) and the DF (Sabour et al., 2015) (k-NN guided) attacks for the targeted and untargeted settings. To evaluate the evasion rate for the *Impersonation Attack*, we adopt the following procedure: first, we randomly select negative matches, and we keep the second image fix, i.e., we analyze pairs of faces $(x,\ x^-)$ where x and x^- belong to different classes. Then, we look upon an adversarial image, whose adversarial class corresponds to the one of x^-, and use it in place of x, i.e., we account the pairs $(x_{adv},\ x^-)$ where x_{adv} is an adversarial example, crafted from x, whose adversarial class is the same as the x^- one. Finally, we consider two similarity measurements:

- "Original", which represents the value of the cosine among the deep features of x and x^-;
- "Adversarial", which represents the cosine between the deep features of x_{adv} and x^-.

Table 2
Percentage of matches which overcome the EER threshold, before (left column) and after (right column) the attacks, considering the targeted and untargeted settings. We highlight in bold the best attack's results.

	Targeted		Untargeted	
	Original	Adv.	Original	Adv.
$\delta = 5$	4.0	92.8	19.9	81.5
$\delta = 7$	4.0	**93.9**	19.9	**81.6**
$\delta = 10$	4.4	93.6	19.8	81.1
CW[a]	50.5	34.9	8.0	9.3

[a]Carlini and Wagner (2017b).

Table 3
Percentage of matches which are below the EER threshold, before (left column) and after (right attack) the attacks, considering the targeted and untargeted settings. We highlight in bold the best attack's results.

	Targeted		Untargeted	
	Original	Adv.	Original	Adv.
$\delta = 5$	5.4	18.9	4.1	9.9
$\delta = 7$	4.3	22.9	4.3	11.4
$\delta = 10$	4.3	**26.2**	4.1	**12.5**
CW[a]	4.0	17.1	4.0	6.2

[a]Carlini and Wagner (2017b).

We report the results in Fig. 7.

In Table 2, we report the percentage of matches which overcome the EER threshold, before and after the attacks, considering the targeted and untargeted settings.

Clearly, from Table 2, the DF attacks (k-NN-guided) are much more effective than CW (Carlini and Wagner, 2017b) in pushing the similarity between the adversarials samples and the original images above the recognition threshold. Such a conclusion holds for targeted and untargeted attacks. Instead, the behavior of the CW (Carlini and Wagner, 2017b) is unpredictable in this setup. Thus we conclude that even though the CW (Carlini and Wagner, 2017b) algorithm is among the strongest attacks concerning the classification objective, it is not very effective against a similarity-based procedure such as the face verification protocol.

We now move to the *Evading Attack* scenario. Differently from the previous case, we start by collecting positive matches, i.e., pairs of images $(x,\ x^+)$ in which x and x^+ belong to the same class, and then we substitute x with one of its adversarial, x_{adv}, whose class is different from the x^+ one. Thus, we obtain the following similarity measurements:

- "Original" which represents the value of the cosine among the deep features of x and x^+;
- "Adversarial" which represents the cosine between the deep features of x_{adv} and x^+.

As mentioned before, the attack's purpose is to push the similarity below the FR system operational level. Fig. 8 and Table 3 show the results.

Table 3 shows the percentage of the matches below the EER threshold, before and after the attacks, considering targeted and untargeted settings.

By observing Table 3, it is clear that the DF attacks (k-NN-guided) are more effective than CW (Carlini and Wagner, 2017b) in this case too. We notice that, on average, the targeted attacks perform better than the untargeted ones, which is expected behavior since an untargeted attack ends as soon as the adversarial is associated with a different identity, therefore it would not have gone any further from the original image.

5.3.1. Detection

Finally, we test our detectors, trained on classification attacks (Section 5.1), on the newly generated adversarial examples. Specifically,

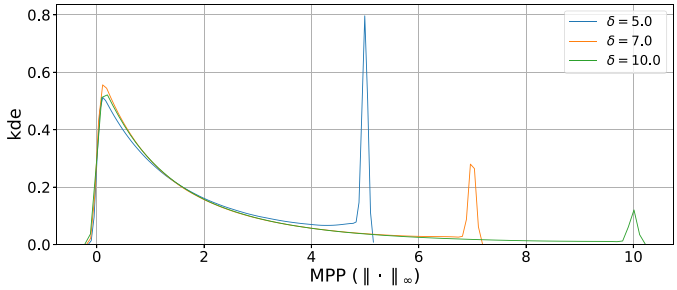

Fig. 6. Maximum Pixel Perturbation (MPP) distribution considering targeted deep representations with different thresholds.

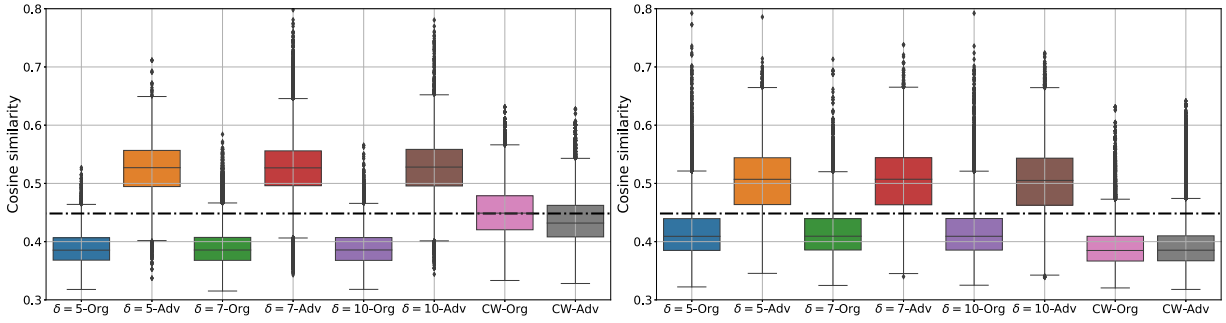

Fig. 7. Cosine similarity distribution for k-NN-guided and CW (Carlini and Wagner, 2017b) attacks in the *Impersonation Attack* scenario. Left: targeted attacks. Right: untargeted attacks. "- Org" refers to the cosine among natural images while "- Adv" refers to the cosine between the natural image and the adversarial one. The dash-pointed line represents the EER threshold.

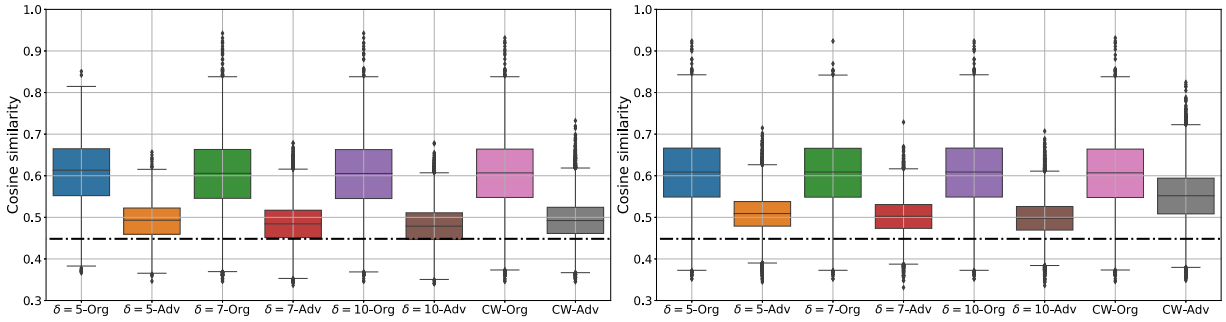

Fig. 8. Cosine similarity distribution for k-NN-guided and CW (Carlini and Wagner, 2017b) attacks in the *Evading Attack* scenario. Left: targeted attacks. Right: untargeted attacks. "- Org" refers to the cosine among natural images while "- Adv" refers to the cosine among the natural image and the adversarial one. The dash-pointed line represents the EER threshold.

Table 4
AUC values for the best performing detector for each threshold value considered in our experiments for both targeted and untargeted settings.

Configuration	AUC		δ
	Targ.	Untarg.	
LSTM + L_2	0.969	0.671	5
LSTM + L_2	0.968	0.688	7
LSTM + L_2	0.972	0.700	10

considering Table 1, we exploit the LSTM detector equipped with the cosine similarity, and we use the class centroids to construct the images' embedding (Section 4). As a summary of our results, we report the AUC values in Table 4 for targeted and untargeted attacks.

According to the results shown in Table 4, we see that, even though the adversarial detector is trained on different attacks, it displays high performance in detecting k-NN-guided ones too. This is a relevant result since it means that, despite the different attacks' objective, adversarial examples share some common behaviors in the inner layers of a deep model that can be exploited to defend against them. Moreover, it highlights the generalization capacity of our detection approach.

5.4. Defense-aware adversary

In the previous sections, we have simulated attacks to an FR system considering an adversary not aware of the detection algorithm. We now change this paradigm, and we give the attacker access to our detector, i.e., he or she is now a *Perfect-Knowledge Adversary* (Biggio et al., 2013; Carlini and Wagner, 2017a) that can use this knowledge to fool both the DCNN and the detection scheme simultaneously.

To this aim, we modify the deep features attack algorithm. We focus our efforts on the targeted scenario: we consider three different values for the maximum allowed perturbation, $\delta = 5.0,\ 7.0,\ 10.0$, and we run each attack for 1000 maximum iterations.

To tamper the images, we consider two settings: a former one, in which the classifier and the detector losses have the same weight, and a latter one, in which we enhance the weight of the detector loss by a factor ten. The difference between the two approaches is that in the second case, the adversarial algorithm focuses more on the detector, thus producing samples in which the manipulation is, in a few cases, slightly perceptible at naked eyes, even though it is still respected the constraint on the maximum perturbation allowed. On the contrary, in the first case, the adversarial examples are always indistinguishable

from the authentic images. We report an example of this behavior in Fig. 9.

In exchange for the loss in imperceptibility, the defense-aware adversary gains a much higher evasion rate. Indeed, considering a maximum perturbation $\delta \in 5.0, 7.0, 10.0$ the success rates in fooling our detection approach are 22.6%, 20.85%, and 18.8% in the former setting and 98.63%, 99.39%, and 99.42% in the latter one, respectively. Interestingly, in the first setting, the higher the allowed pixel perturbation δ, the lower the evasion success. Such a trade-off can be explained, considering how our detection approach works. A more significant perturbation in the pixel space corresponds to a larger difference in the deep representations. Since our detection criterion looks for different behaviors in the features space, it might be the case that more perturbed images are easily spotted as adversarial from our detection algorithm compared to less perturbed ones. On the contrary, we did not observe such behavior in the second setting since the attacker is now more focused on fooling the detection process.

From our results, it is evident that the defense-aware adversary setting still poses challenges that will drive future researches. Moreover, we notice a significant resilience of our system against white-box attacks, compelling the attacker to pay a small price in terms of imperceptibility in the attempt to fool the detector.

5.5. Benchmark datasets attacks detection

Even though our focus is on FR systems, we dedicate this section to test our detection algorithm on smaller commonly adopted datasets, to compare our approach to others available in the literature. To this aim, we use the MNIST (LeCun et al., 1998a,b) and CIFAR-10 (Krizhevsky et al., 2009a,b) datasets. Concerning the threatened models, we adopt a LeNet-like (Madry et al., 2017) architecture for the former set and the WideResNet (Zagoruyko and Komodakis, 2016) for the latter one. We train both models with the SGD optimizer and an initial learning rate of 0.1 that we drop by a factor ten every 50 epochs. We finally obtain a classification accuracy on the test set of 99.5% and of 95.7% for the LeNet-like (Madry et al., 2017) and the WideResNet (Zagoruyko and Komodakis, 2016), respectively.

Concerning the adversarial examples, we use the test set of the two datasets as data source. To detect the manipulated images, we use our LSTM detector equipped with the L_2 metric and with medoids as class reference points. To craft the malicious samples, we use the PGD (Madry et al., 2017) and the MI-FGSM (Dong et al., 2018) attacks with the L_∞ norm, and the BIM (Kurakin et al., 2016a) and the CW Carlini and Wagner (2017b) attacks with the L_2 norm. To compare with the results available in the literature, we generate adversarial examples using the most commonly adopted settings. Specifically, concerning the first three attacks, we generate samples with $\epsilon \in \{0.1, 0.3, 0.5\}$ for MNIST (LeCun et al., 1998a,b) and with $\epsilon \in \{0.03, 0.07, 0.1\}$ for CIFAR-10 (Krizhevsky et al., 2009a,b), and we run the attacks for maximum number of iterations in $\{10, 40, 50, 100\}$ for both datasets. Concerning the CW (Carlini and Wagner, 2017b) attack, we run it with binary search steps in $\{5, 10, 30, 50\}$ and maximum iterations in $\{10, 100, 1000\}$. For all of the attacks, we manipulate images considering the targeted and the untargeted settings and then merge them into a unified set.

We report our results in Tables 5 and 6 for the MNIST (LeCun et al., 1998a,b) and the CIFAR-10 (Krizhevsky et al., 2009a,b) datasets, respectively. When available, we also report the results from other detection techniques. Specifically, we report the values relative to the attacks' settings that best agree with ours.

As we can appreciate from Tables 5 and 6, our method reaches the highest performance compared to the other detection approaches in almost all the cases.

Lastly, we compare the performance of our method with the Deep Neural Rejection (DNR) approach proposed by Sotgiu et al. (2020). For this purpose, we use the code that the authors made publicly available in the *secml*.[4] library and we train the SVM classifiers on top of the same DCNN used in Table 5 The DNR technique shares with our approach the idea of leveraging the features extracted from different layers of a deep model under attack to reject adversarial instances. Specifically, we apply the DNR technique considering the last three layers of the DCNN (we do not observe improvements in terms of performance by using more layers). Moreover, to evaluate the performance of the two detection approaches, we consider the same adversarial instances used for the evaluation in Table 5. Differently from the previous results, we do not report the AUC for each attack. Instead, we evaluate the rejection accuracy at a given value of the false positive rejection (fpr) rate. Thus, we exploit a protocol similar to what Sotgiu et al. (2020) proposed in their work. We report the results in Table 7.

As shown in Table 7, also in this case our detection technique reaches the highest performance in terms of adversarial detection at a given fpr value.

Finally, we conduct defense-aware attacks against our detection system. To this aim, we employ the CW (Carlini and Wagner, 2017b) attack with the L_2 norm, and we set 10 binary search steps and 1000 maximum iterations. As we have reported in Section 5.4, to obtain a higher evasion rate, we use a different weight between the DL model and the detector losses in this case too. Specifically, we obtain the highest evasion rates multiplying the latter by a factor 1000 compared to the former one. We report our results, along with others available in literature in Table 8.

From Table 8, it is evident that our detection approach is much more resilient to adversarial attacks in the defense-aware setting than other proposed procedures.

Table 5
Detection rate on MNIST (LeCun et al., 1998a,b) adversarials. We emphasize in bold the performance of the best model.

Model	PGD	BIM	CW	MI-FGSM
Feinman et al. (2017)	–	97.2	97.9	–
Ma et al. (2018)[a]	90.2	–	100.	–
Lee et al. (2018)[a]	73.6	–	95.7	–
Yang et al. (2019)	100.	–	**100.**	–
Our method	**100.**	**99.8**	99.7	**100.**

[a]Values reported in Yang et al. (2019).

Table 6
Detection rate on CIFAR-10 (Krizhevsky et al., 2009a,b) adversarials. We emphasize in bold the performance of the best model.

Model	PGD	BIM	CW	MI-FGSM
Feinman et al. (2017)	–	81.1	92.2	–
Ma et al. (2018)[a]	81.2	–	44.5	–
Lee et al. (2018)[a]	81.3	–	64.1	–
Yang et al. (2019)	95.3	–	71.1	–
Our method	**99.2**	**98.5**	**98.4**	**99.3**

[a]Values reported in Yang et al. (2019).

Table 7
Detection rate on MNIST (LeCun et al., 1998a,b) adversarials. We emphasize in bold the performance of the best model.

fpr	PGD		BIM		CW		MI-FGSM	
	DNR[a]	Our	DNR[a]	Our	DNR[a]	Our	DNR[a]	Our
0.01	73.2	**99.9**	69.7	**96.0**	91.3	**94.5**	78.5	**99.8**
0.03	89.1	**99.9**	85.1	**99.0**	99.0	**99.9**	91.6	**99.9**
0.05	91.5	**99.9**	88.5	**99.5**	99.8	**99.9**	93.7	**99.9**
0.10	94.4	**99.9**	91.6	**99.7**	**100.**	99.9	95.1	**99.9**

[a]Sotgiu et al. (2020).

[4] https://gitlab.com/secml/secml/-/tree/master.

Fig. 9. Left: source image and adversarial example crafted with similarity and detector losses equally balanced. Right: adversarial example generated with a higher weight for the detector loss. In the malicious sample, the distortion is slightly perceivable.

Table 8
Defense-aware attacks success rate for the MNIST (LeCun et al., 1998a,a) dataset. We emphasize in bold the performance of the best model.

Method	Evasion rate (%)
Feinman et al. (2017)[a]	98.0
Gong et al. (2017)[a]	100.
Grosse et al. (2017)[a]	100.
Liang et al. (2018)	67.4
Our method (targeted)	**8.3**
Our method (untargeted)	**11.4**

[a]Values reported in Liang et al. (2018)

6. Conclusions

Adversarial examples represent a severe threat to DL models, as they set a considerable limitation, especially on the use of learning models in sensitive applications. Despite the scientific community's effort to train robust NNs, a knowledgeable attacker usually succeeds in finding ways to attack a model.

Except for the adversarial training, another approach to enhance the robustness of AI-based systems to the adversarial threat is detecting these malicious inputs. In several previous studies, the properties of the offensive samples are exploited to detect them. Compared to adversarially training a model, the detection of these images has several advantages, e.g., it does not require to re-train any model, nor does it not need to specially design new training strategies to flatten the model loss manifold.

In light of these facts, we proposed our study on the detection of the adversarial examples. Specifically, we exploit their different behavior in the inner layers of a DL model concerning natural images.

We conducted our experiments in the context of Face Recognition (FR), for which we crafted adversarial examples considering wrong-label assignment and deep representation distance as objectives in the targeted and untargeted settings. We first consider the DNN acting as a classifier, and then we conducted our attacks against an FR system in which the learning model is employed as a features extractor. As far as the classification attacks are concerned, the best detector reaches an AUC value of 99% on the adversarial detection task.

The results obtained from the deep features attacks against an FR system are even more impressive. In this case, we considered a more realistic application scenario for an FR system in which the DL model was used as a features extractor, and the final task was accomplished employing similarity measurements among the descriptors vectors. Specifically, first, we observed that classification attacks are much less effective in fooling an FR system. Second, the detectors, trained on the first type of attacks, reached an AUC value of 97% and 70% for the deep representation attacks, which they had never seen before, for targeted and untargeted attacks, respectively. These last results are of a significant impact considering the idea of an "universal" adversarial detector. Moreover, this also means that, despite the different objectives of the various kind of attacks, they share some common properties that can, or perhaps should, be exploited to recognize adversarial attacks and build more robust systems without the need to change the model to increase its robustness periodically. Third, we observed that our detection algorithm showed a high degree of resilience against defense-aware adversaries.

Finally, we compared our detection approaches with others available in the literature. Concerning the comparisons, our method reached the highest detection performance, and it showed itself to be the most resilient against defense-aware attacks.

CRediT authorship contribution statement

Fabio Valerio Massoli: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization. **Fabio Carrara:** Conceptualization, Software, Writing - original draft, Writing - review & editing. **Giuseppe Amato:** Project administration, Funding acquisition, Review. **Fabrizio Falchi:** Conceptualization, Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the AI4EU project, funded by the EC (H2020 - Contract n. 825619), and by Automatic Data and documents Analysis to enhance human-based processes (ADA) project, CUP CIPE D55F17000290009. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

References

Amato, G., Carrara, F., Falchi, F., Gennaro, C., Vairo, C., 2018. Facial-based intrusion detection system with deep learning in embedded devices. In: Proceedings of the 2018 International Conference on Sensors, Signal and Image Processing. ACM, pp. 64–68.

Amirian, M., Schwenker, F., Stadelmann, T., 2018. Trace and detect adversarial attacks on CNNs using feature response maps. In: 8th IAPR TC3 Workshop on Artificial Neural Networks in Pattern Recognition. ANNPR, Siena, Italy, September 19–21, 2018, IAPR.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F., 2013. Evasion attacks against machine learning at test time. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 387–402.

Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A., 2018. Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition. FG 2018, IEEE, pp. 67–74.

Carlini, N., Wagner, D., 2017a. Adversarial examples are not easily detected: Bypassing ten detection methods. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM, pp. 3–14.

Carlini, N., Wagner, D., 2017b. Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy. SP, IEEE, pp. 39–57.

Carrara, F., Becarelli, R., Caldelli, R., Falchi, F., Amato, G., 2018b. Adversarial examples detection in features distance spaces. In: Proceedings of the European Conference on Computer Vision, ECCV.

Carrara, F., Esuli, A., Fagni, T., Falchi, F., Moreo Fernández, A., 2018a. Picture it in your mind: generating high level visual representations from textual descriptions. Inf. Retr. J. (ISSN: 1573-7659) 21 (2), 208–229. http://dx.doi.org/10.1007/s10791-017-9318-6.

Carrara, F., Falchi, F., Caldelli, R., Amato, G., Becarelli, R., 2019. Adversarial image detection in deep neural networks. Multimedia Tools Appl. 78 (3), 2815–2835.
Deng, L., Liu, Y., 2018. Deep Learning in Natural Language Processing. Springer.
Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J., 2018. Boosting adversarial attacks with momentum. arXiv preprint.
Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., Zhu, J., 2019. Efficient decision-based black-box adversarial attacks on face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7714–7722.
Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B., 2017. Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410.
Feldstein, S., 2019. The global expansion of AI surveillance. Working Paper. Carnegie Endowment for International Peace. 1779 Massachusetts Avenue NW, Washington, DC 20036. URL https://carnegieendowment.org/files/WP-Feldstein-AISurveillance_final1.pdf.
Girshick, R., 2015. Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448.
Gong, Z., Wang, W., Ku, W.-S., 2017. Adversarial and clean data are not twins. arXiv preprint arXiv:1704.04960.
Goodfellow, I.J., Shlens, J., Szegedy, C., 2014. Explaining and harnessing adversarial examples (2014). arXiv preprint arXiv:1412.6572.
Goswami, G., Agarwal, A., Ratha, N., Singh, R., Vatsa, M., 2019. Detecting and mitigating adversarial perturbations for robust face recognition. Int. J. Comput. Vis. 127 (6–7), 719–742.
Grosse, K., Manoharan, P., Papernot, N., Backes, M., McDaniel, P., 2017. On the (statistical) detection of adversarial examples. arXiv preprint arXiv:1702.06280.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.
Huang, R., Xu, B., Schuurmans, D., Szepesvári, C., 2015. Learning with a strong adversary. arXiv 2015. arXiv preprint arXiv:1511.03034.
Kakizaki, K., Yoshida, K., 2019. Adversarial image translation: Unrestricted adversarial examples in face recognition systems.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Krizhevsky, A., Hinton, G., et al., 2009a. Learning Multiple Layers of Features from Tiny Images. Citeseer.
Krizhevsky, A., Hinton, G., et al., 2009b. The CIFAR-10 database. URL https://www.cs.toronto.edu/~kriz/cifar.html.
Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, Vol. 25. Curran Associates, Inc., pp. 1097–1105, URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.
Kurakin, A., Goodfellow, I., Bengio, S., 2016a. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533.
Kurakin, A., Goodfellow, I., Bengio, S., 2016b. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533.
LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998a. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.
LeCun, Y., et al., 1998b. The MNIST database of handwritten digits. URL http://yann.lecun.com/exdb/mnist.
Lee, K., Lee, K., Lee, H., Shin, J., 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: Advances in Neural Information Processing Systems. pp. 7167–7177.
Li, X., Li, F., 2017. Adversarial examples detection in deep networks with convolutional filter statistics. In: ICCV. pp. 5775–5783.
Liang, B., Li, H., Su, M., Li, X., Shi, W., Wang, X., 2018. Detecting adversarial image examples in deep neural networks with adaptive noise reduction. IEEE Trans. Dependable Secure Comput..
Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., Zhu, J., 2018. Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1778–1787.
Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L., 2017. Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 212–220.
Lu, J., Issaranon, T., Forsyth, D., 2017. Safetynet: Detecting and rejecting adversarial examples robustly. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 446–454.
Ma, X., Li, B., Wang, Y., Erfani, S.M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M.E., Bailey, J., 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. arXiv preprint arXiv:1801.02613.
Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2017. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.
Massoli, F.V., Amato, G., Falchi, F., 2020. Cross-resolution learning for face recognition. Image Vis. Comput. 103927.
Massoli, F.V., Amato, G., Falchi, F., Gennaro, C., Vairo, C., 2019. Improving multi-scale face recognition using VGGFace2. In: International Conference on Image Analysis and Processing. Springer, pp. 21–29.
Maze, B., Adams, J., Duncan, J.A., Kalka, N., Miller, T., Otto, C., Jain, A.K., Niggel, W.T., Anderson, J., Cheney, J., et al., 2018. IARPA janus benchmark-c: Face dataset and protocol. In: 2018 International Conference on Biometrics. ICB, IEEE, pp. 158–165.
Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B., 2017. On detecting adversarial perturbations. arXiv preprint arXiv:1702.04267.
Ortis, A., Farinella, G.M., Battiato, S., 2019. An overview on image sentiment analysis: Methods, datasets and current challenges. In: Proceedings of the 16th International Joint Conference on E-Business and Telecommunications, ICETE 2019 - Vol. 1. DCNET, ICE-B, OPTICS, SIGMAP and WINSYS, Prague, Czech Republic, July 26-28, 2019, pp. 296–306. http://dx.doi.org/10.5220/0007909602900300.
Papernot, N., McDaniel, P., 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. arXiv preprint arXiv:1803.04765.
Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A., 2015. Distillation as a defense to adversarial perturbations against deep neural networks. arXiv preprint arXiv:1511.04508.
Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A., 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy. SP, IEEE, pp. 582–597.
Qiu, H., Xiao, C., Yang, L., Yan, X., Lee, H., Li, B., 2019. Semanticadv: Generating adversarial examples via attribute-conditional image editing. arXiv preprint arXiv:1906.07927.
Sabour, S., Cao, Y., Faghri, F., Fleet, D.J., 2015. Adversarial manipulation of deep representations. arXiv preprint arXiv:1511.05122.
Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K., 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 1528–1540.
Sitawarin, C., Wagner, D., 2019. On the robustness of deep K-nearest neighbors. arXiv preprint arXiv:1903.08333.
Song, Q., Wu, Y., Yang, L., 2018. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network. arXiv preprint arXiv:1811.12026.
Sotgiu, A., Demontis, A., Melis, M., Biggio, B., Fumera, G., Feng, X., Roli, F., 2020. Deep neural rejection against adversarial examples. EURASIP J. Inf. Secur. 2020 (1), 1–10.
Sundararajan, K., Woodard, D.L., 2018. Deep learning for biometrics: a survey. ACM Comput. Surv. 51 (3), 65.
Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.
Turk, M.A., Pentland, A.P., 1991. Face recognition using eigenfaces. In: Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 586–591.
Wang, M., Deng, W., 2018. Deep face recognition: A survey. arXiv preprint arXiv:1804.06655.
Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., Kalka, N., Jain, A.K., Duncan, J.A., Allen, K., et al., 2017. Iarpa janus benchmark-b face dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 90–98.
Yan, Z., Guo, Y., Zhang, C., 2018. Deep defense: Training dnns with improved adversarial robustness. In: Advances in Neural Information Processing Systems. pp. 419–428.
Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L., Jordan, M.I., 2019. ML-LOO: Detecting adversarial examples with feature attribution. arXiv preprint arXiv:1906.03499.
Zagoruyko, S., Komodakis, N., 2016. Wide residual networks. arXiv preprint arXiv:1605.07146.