# Puzzling Face Verification Algorithms for Privacy Protection

Binod Bhattarai, Alexis Mignon, Frédéric Jurie, Teddy Furon

▶ **To cite this version:**

Binod Bhattarai, Alexis Mignon, Frédéric Jurie, Teddy Furon. Puzzling Face Verification Algorithms for Privacy Protection. IEEE Workshop on Information Forensics and Security, Dec 2014, Atlanta, United States. hal-01066070

**HAL Id: hal-01066070**

**https://inria.hal.science/hal-01066070**

Submitted on 5 Dec 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Puzzling Face Verification Algorithms for Privacy Protection

Binod Bhattarai*, Alexis Mignon*, Frédéric Jurie*, Teddy Furon†

* *GREYC, Université of Caen, Caen, France*
† *Inria, Rennes, France*

***Abstract*—This paper presents a new approach for de-identifying face images, *i.e.* for preventing automatic matching with public face collections. The overall motivation is to offer tools for privacy protection on social networks. We address this question by drawing a parallel between face de-identification and oracle attacks in digital watermarking. In our case, the identity of the face is seen as the watermark to be removed. Inspired by oracle attacks, we forge de-identified faces by superimposing a collection of carefully designed noise patterns onto the original face. The modification of the image is controlled to minimize the probability of good recognition while minimizing the distortion. In addition, these de-identified images are – by construction – made robust to counter attacks such as blurring. We present an experimental validation in which we de-identify LFW faces and show that resulting images are still recognized by human beings while deceiving a state-of-the-art face recognition algorithm.**

## I. Introduction

Posting photos of oneself and relatives is one of the main activity on social networks. Yet, these are pictures with visible faces, and faces are distinctive. Thanks to an automatic face recognition solution (and all the major actors in the field have recently acquired such technology), the network proposes the user to cross link faces on his pictures with profiles of his acquaintances. This functionality, called "Photo Tag Suggest" on Facebook for instance, uses already labeled photos and face recognition technology to identify individuals in new photos.

If Yana Welinder doesn't lower the fantastic innovation behind social networks, she points the numerous breaches of individual privacy bound to sharing face pictures on this medium [20]. Face recognition technology in conjunction with social networks shifts the anonymity paradigm. A priori anonymous faces are not only connected to names but also to all the information of social network profiles. Welinder shows that law alone cannot prevent these dangers, she stresses that it is up to the user to decide to benefit from this functionality or not, and that this user-centric privacy policy should be enforced together by legal and technological means.

This appeal for a technological privacy gatekeeper motivates our work. More precisely, we are investigating whether a user could post a picture with her/his face publicly visible on the social network such that friends recognize him, while, at the same time, the automatic face tagging technology fails.

Fig. 1. Pairs of images from LFW. Left images are original images and right images are de-identified forgeries.

Face tagging is usually done thanks to a face verification algorithm. To compare two faces, this algorithm encodes them into discriminative signatures, computes a 'distance' between the signatures, and compares this metric to a threshold. Website [14] provides a benchmark of recent face verification algorithms. In this context, de-identifying an image consists in altering it in such a way that its signature becomes different enough from the signature of a reference image.

Anonymizing faces is easy by masking or blurring them (*e.g.* Google Street View) if the goal is to make them non recognizable by humans. When the image quality has to be preserved (*i.e.* let a face looks like a face), the wording 'de-identification' is preferred to 'anonymization' since a human still recognize a relative on the picture. Better image processing than blurring have been proposed [5], [16]. Section II reviews these works and outlines three pitfalls. They address the sanitization of a database of face images before publication, whereas we de-identify a query image. Their very specific approach, *i.e.* a retro-engineering of the well-known eigenfaces representation, does not apply on modern face recognition schemes, which are much more non-linear. Their experimental work uses images from 'biometric' datasets which are quite different from face images published on social network.

The parallel between our de-identification scenario and oracle attacks in of digital watermarking [3], [4], [7] motivates our approach. A digital watermark is a perceptually invisible marker embedded in multimedia contents. An attack refers to as an image processing partially removing the watermark in the sense that the detector no longer classifies the altered images as watermarked. In some applications, the pirate has access to the watermark detector as an oracle: the pirate has no

knowledge about the watermarking technology ; the watermark detector is a black box, to which the pirate submits images and observes the binary decisions (presence or absence of a watermark). Oracle attacks benefit from this feedback to iteratively refine the quality of the attacked images.

Our problem is similar in the sense that the identifiability of a face is the equivalent of the detectability of a watermark to be hindered. This parallel opens the door to interesting avenues for face de-identification. First, the assumption of an oracle is valid: more and more face recognition tools are publicly available (*e.g.* Google Picasa or Apple iPhoto softwares provide this functionality), the user a priori knows the photos of himself present in the social network used as a reference to identify him. Second, the user doesn't need to know all the internals of the technology, the oracle attack only needs the output of this black box: from an image with a face, the person is correctly identified or not.

Even guided by the feedback of an oracle, the strategy for altering the face image is of utmost importance. The previous approach deeply distorts the images to a point where recognition by humans is challenging (see figures in [5], [16]). On the other hand, hindering the recognition by a computer but not by a human is a well-known CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) ill-posed problem. The keystone idea is that human recognition is more robust than computer recognition against certain types of distortion. Geometrical distortion is widespread in text-based CAPTCHA, but it produces too many unpleasant artefacts when applied on face images. Our approach relies on background noise distortion and the ability of the human brain for sources separation. Seeing a grid on a face image, the human brain spots the two 'sources', the grid and the face, that it easily separates.

Consequently, one key idea of the paper is to combine face images with procedural noise textures (*e.g.* Perlin's noise [17]), with the rational that (i) the stationarity of the noise texture lets human brains remove it, (ii) the parameters of the texture allows to minimize the alteration. An oracle attack controls the trade-off between image quality and de-identification. We also investigate the robustness against counter-attacks. Indeed, the added noise should be robust to filtering otherwise removing it will be easy. This is explicitly taken into account in the optimization of the texture parameters.

The proposed approach is experimentally validated on the popular Labeled Faces on Wild (LFW) dataset [10]. This dataset is close to the targeted use case. Qualitative experiments present altered images for visual inspection. Quantitative experiments evaluate the performance of our method against recent best performing face recognition algorithms. We show that the accuracy of the face recognition system can be reduced from a state-of-the-art 85% (unsupervised face recognition, see [11]) to the level of chance while the alteration of images is acceptable for humans.

## II. Related Work

### A. Face recognition literature

Face de-identification has received very little attention in the computer vision literature. Newton *et al.* [16] propose a privacy enhancing algorithm, called $k$-Same, transposing the concept of $k$-anonymity to face image databases. The algorithm determines similarity between faces of the database, clusters similar faces, and creates a new face by aggregating the faces of a cluster. Gross *et al.* [8], [9] propose a factorization approach to separate identity and non-identity related factors, allowing to only replace the identity-factors by the cluster's aggregation, while keeping the non-identity factors untouched to better preserve facial expressions. Dufaux and Ebrahimi [6] presents an effective scrambling techniques. Recently, Driessen and Dürmuth [5] put the preservation of the human recognition as a top requirement. The idea is to find the modification of the image which has the lowest distortion while changing the signature to a desired value, *i.e.* the aggregation of the cluster's signatures. In practice, they work with the signature extraction based on the face image projection onto the manifold spanned by some eigenfaces. Modifying the signature amounts to change this projection which is easy thanks to the linearity of the process.

We outline that we do not target the same goal. In their context, the owner of a database of pictures would like to publish it with the guarantee that it can not be used for identifying people In other words, their goal is to sanitize the database before publishing it. In our context, the social network has already published a collection of pictures, and it is up to the user to de-identify his new picture. In other words, we manipulate the query image not the database images.

These previous works are dedicated to the well-known eigenfaces representation [16], [5], or some variants [9], which is somehow outdated. Such retro-engineering is much more difficult with recent extraction processes which are highly non-linear [11], [15], [19]. Our approach based on oracle attacks does not rely on retro-engineering the signature extraction. The quality of the forged images is not so good: they look blurred in [16, Fig. 17], and adding some fraction of the eigenfaces result in artifacts like ghosting and rebound effects around the face traits [5, Fig. 9]. None of the previous works addresses the privacy of faces 'in the wild' *i.e.* as they are on the internet. They are focused on datasets like FERET [18] or a subset of CSU multi-pie [1], where images are taken in a controlled environment (frontal illumination, frontal pose, *etc.*).

### B. Digital watermarking literature

Our work is inspired by oracle attacks against digital watermarking. The detection of the presence (or the absence) of an invisible watermark in an image is sufficient in many applications. The embedding and the detection are algorithms relying on a secret key. In some application, it is assumed that the pirate has a free access to a watermark detector in a black sealed box.

Let us consider images as points in the image space $\mathcal{I}$, and denote $\boldsymbol{x}_o$ ($\boldsymbol{x}_w$) the original image (resp. its watermarked

version). The detection function $D : \mathcal{I} \to \{0,1\}$ outputs 1 if the watermark is deemed present. We call $\mathcal{D} = \{\boldsymbol{z} \in \mathcal{I} | D(\boldsymbol{z}) = 1\}$ the set of images deemed as watermarked and $\mathcal{B}$ the boundary between $\mathcal{D}$ and $\bar{\mathcal{D}}$. The attacker aims at finding an image $\boldsymbol{y}$ such that $D(\boldsymbol{y}) = 0$ and as close as possible to $\boldsymbol{x}_w$. The distortion is measured by function $d(\boldsymbol{x}_w, \boldsymbol{y})$, say the Euclidean distance. Formally, the optimal attack can be written as: the following optimization problem:

$$\boldsymbol{y}^\star = \arg \min_{\boldsymbol{y}: D(\boldsymbol{y})=0} d(\boldsymbol{x}_w, \boldsymbol{y}) \tag{1}$$

Comesaña *et al.* [3], showed that (1) is equivalent to:

$$\boldsymbol{y}^\star = \arg \min_{\boldsymbol{y}} \delta(\boldsymbol{y}) \tag{2}$$

where $\delta(\boldsymbol{y}) = d(\boldsymbol{x}_w, h_{\mathcal{B}}(\boldsymbol{y}))$ and $h_{\mathcal{B}}(\cdot) : \mathcal{I} \to \mathcal{B}$ is a surjection onto the boundary of the detection region. The shape and location of this boundary depends on the secret key. This prevents the pirate to implement it. Yet, thanks to the free access of the watermark detector, oracle attacks aim at solving this minimization problem. Several implementations have been proposed. They first differ by the implementation of the surjection. For example, when $\boldsymbol{y}$ is such that $D(\boldsymbol{y}) = 0$, a classical trick is to perform a line search over $\alpha \in (0,1)$ such that $\hat{h}_{\mathcal{B}}(\boldsymbol{y}) = \boldsymbol{x}_w + \alpha(\boldsymbol{y} - \boldsymbol{x}_w)$ is close to $\mathcal{B}$. Another difference is the way the oracle attack locally explores $\mathcal{B}$: Comesaña estimates the gradient (or also the Hessian) of the function $\delta(\boldsymbol{y})$, which costs $O(N)$ (resp. $O(N^2)$) oracle calls, in order to perform a Newton-Raphson method [3], [4]. To save oracle calls, Earl [7] keeps on randomly drawing a new direction in the space and tests whether moving along this direction may decrease the functional $\delta(\boldsymbol{y})$.

An oracle attack thus 'travels' over the boundary until it finds a minimum of $\delta(\boldsymbol{y})$. This is a local minimum because the detection region is a priori not convex. It drastically reduces the average pixel distortion (around $10^{-4}$, *i.e.* PSNR of 40dB) required for removing the watermark compared to blind attacks like a coarse JPEG compression (around $10^{-2}$, *i.e.* PSNR of 20dB). The main criterion to compare oracle attacks is the number of calls to the watermarking detector.

## III. Our method

We consider a given face verification algorithm determining if a face image represents the same person as a reference image $\boldsymbol{x}_o$. We model it as a binary function of the image space: $V_{\boldsymbol{x}_o}(\cdot) : \mathcal{I} \to \{0,1\}$. We start from a face image $\boldsymbol{x}_f$ representing the same person that the system succeeds to identify: $\boldsymbol{x}_o$ (*i.e.* $V_{\boldsymbol{x}_o}(\boldsymbol{x}_f) = 1$). De-identifying consists in forging a new image $\boldsymbol{y}$ such that $V_{\boldsymbol{x}_o}(\boldsymbol{y}) = 0$ and $d(\boldsymbol{x}_f, \boldsymbol{y})$, the distortion metric between $\boldsymbol{x}_f$ and $\boldsymbol{y}$, is small.

### A. Oracle attack for face de-identification

We transpose the oracle attack to the field of face recognition by replacing the watermark detector $D(\cdot) : \mathcal{I} \to \{0,1\}$ by the face verification function $V_{\boldsymbol{x}_o}(\cdot) : \mathcal{I} \to \{0,1\}$. In the same way, we define $\mathcal{V}_{\boldsymbol{x}_o} = \{\boldsymbol{x} | V_{\boldsymbol{x}_o}(\boldsymbol{x}) = 1\}$ as the set of images detected as representing the person as in $\boldsymbol{x}_o$, and $\mathcal{B}_{\boldsymbol{x}_o}$ the boundary between $\mathcal{V}_{\boldsymbol{x}_o}$ and $\overline{\mathcal{V}_{\boldsymbol{x}_o}}$. We aim at optimizing the following problem:

$$\begin{aligned} \boldsymbol{y}^\star &= \arg \min_{y: V_{\boldsymbol{x}_o}(\boldsymbol{y})=0} d(\boldsymbol{x}_f, \boldsymbol{y}) && (3) \\ &= \arg \min d(\boldsymbol{x}_f, h_{\mathcal{B}_{\boldsymbol{x}_o}}(\boldsymbol{y})), && (4) \end{aligned}$$

where $h_{\mathcal{B}_{\boldsymbol{x}_o}}(\boldsymbol{y})$ is a surjection onto the boundary $\mathcal{B}_{\boldsymbol{x}_o}$.

### B. Main ingredients

We work with the approach of Earl rather than the method of Comesaña to make less oracle calls (see Sect. II-B). We present the exploration of $\mathcal{B}_{\boldsymbol{x}_o}$ by the synthesis of noise textures, the approximate surjection, and the main algorithm.

*1) Texture parametrization:* Let $\mathcal{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_N\}$ be a set of $N$ texture images chosen for their statistical or visual properties detailed later on. The images have pixel values in the range $[0, 255]$. The noise $\boldsymbol{t}$ is computed as a linear combination of the textures of $\mathcal{T}$. Writing the images column wise, $T = [\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_N]$ is the matrix where the $i$-th column corresponds to the $i$-th texture image, and:

$$\boldsymbol{t} = \sum_{i=1}^{N} \beta_i \boldsymbol{t}_i = T\boldsymbol{\beta} \tag{5}$$

where $\boldsymbol{\beta}$ contains the coefficients of the combination. To ensure that the pixels of $\boldsymbol{t}$ are in $[0, 255]$, we add the constraints: $\beta_i \in [0,1]$, $\forall i$ and $\sum_i \beta_i = 1$. In other words, $\boldsymbol{\beta}$ lies in the standard $(N-1)$-simplex of $\mathbb{R}^N$, denoted by $\Delta^{N-1}$.

*2) The surjection function:* In our method, the forgery $\boldsymbol{y}$ is computed as the interpolation between the original image $\boldsymbol{x}_f$ and a noise image $\boldsymbol{t}$: $\boldsymbol{y} = (1-\alpha)\boldsymbol{x}_f + \alpha\boldsymbol{t}$. The distortion function $d(\boldsymbol{x}_f, \boldsymbol{y}) = \|\boldsymbol{x}_f - \boldsymbol{y}\|^2$ becomes:

$$d(\boldsymbol{x}_f, \boldsymbol{y}) = \alpha^2 \|T\boldsymbol{\beta} - \boldsymbol{x}_f\|^2. \tag{6}$$

Since $V_{\boldsymbol{x}_o}(\boldsymbol{x}_f) = 1$ and $V_{\boldsymbol{x}_o}(\boldsymbol{t}) = 0$, there exists a value $\alpha(\boldsymbol{\beta}) \in (0,1]$ which brings $\boldsymbol{y}$ on the boundary $\mathcal{B}_{\boldsymbol{x}_o}$. The surjection consists in finding this appropriate value. In practice, this value is approximated by a bisection as in [4]. With these notations, we aim at solving the following problem:

$$\min_{\boldsymbol{\beta} \in \Delta^{N-1}, V_{\boldsymbol{x}_o}(\boldsymbol{x}_f + \alpha(\boldsymbol{\beta})(T\boldsymbol{\beta} - \boldsymbol{x}_f)) = 0} \alpha(\boldsymbol{\beta})^2 \|T\boldsymbol{\beta} - \boldsymbol{x}_f\|^2 \tag{7}$$

*3) Optimization over the standard $(N-1)$-simplex:* The region $\mathcal{V}_{\boldsymbol{x}_o}$ is a priori not a convex set, whence neither $\alpha(\boldsymbol{\beta})$ nor the functional to be minimized in (7) are convex functions. This makes our problem difficult to solve exactly. Again, we follow the same path as Earl in [7] by resorting to a stochastic approximate optimization, detailed in Algorithm 1. This is a stochastic variant of coordinate descent and convergence guarantees can be given when the functional is convex [12], [13]. Again, this doesn't hold in our case, but we do observe a convergence to (likely) local minima. This shows that, if region $\mathcal{V}_{\boldsymbol{x}_o}$ may not be convex, it is certainly piecewise convex.

**Algorithm 1** Greedy approximation over standard simplex

1: **procedure** GREEDYSMPLX($f$)
2: **Input:** a function $f(\boldsymbol{\beta})$, $\{\boldsymbol{e}_i\}$ vertices of the simplex
3: **Ouput:** an approximate solution $\boldsymbol{\beta}^*$
4: $k \leftarrow 0$
5: $i_0 \leftarrow random(N)$
6: $\boldsymbol{\beta}^0 \leftarrow \boldsymbol{e}_{i_0}$
7: **while** not converged **do**
8: $k \leftarrow k+1$
9: $i_k \leftarrow random(N)$
10: $\eta_k \leftarrow linesearch(f(\boldsymbol{\beta}^{k-1} + \eta(\boldsymbol{e}_{i_k} - \boldsymbol{\beta}^{k-1})))$
11: $\boldsymbol{\beta}^k \leftarrow \boldsymbol{\beta}^{k-1} + \eta_k(\boldsymbol{e}_{i_k} - \boldsymbol{\beta}^{k-1})$
12: **end while**
13: **return** $\boldsymbol{\beta}^k$
14: **end procedure**

## IV. EXPERIMENTS

The overall approach for validation is threefold. (1) very strict experiments are first performed in a so-called *self de-identification* setting. The objective is to forge face images, which, when compared to themselves (forged vs original image), are considered by the face recognition algorithm as representing two different persons. We compare the proposed approach to de-identification by blurring or jpeg compression. (2) A method for making our de-identification more robust to simple counter attacks, is proposed, here again within the same self de-identification context. (3) finally, a more realistic (while easier) set of experiments is proposed. The goal is to de-identify positive face pairs (*i.e.* pairs of different images representing the same person) by altering one of the two images, the other being considered as the reference image. In this context, the recognition rate of human subjects on de-identified images is also evaluated.

All the experimental validations are done with the Labeled Faces in the Wild (LFW) face database [10]. The choice of this database is led by the great variability of faces with respect to the pose, lighting and expression conditions compared to other popular databases like FERET [18]. Faces contained in this dataset are very close to the application context we are interested in, *i.e.* the privacy of face images on social networks.

Regarding face encoding, the I-LQP descriptor [11] is used. While recent methods [2], [19] give slightly better performances, I-LQP has the great advantage of being much faster to compute: about 10 times faster than the Fisher Vector based encoder of [19]. This is an important issue since any call to the oracle implies the signature extraction from a new image.

Finally, we use three types of image noise to alter the image. Since most modern face recognition algorithms rely on statistics of local features, textures good at de-identification should exhibit energetic components at the same scale as characteristic faces features. Another point is that those textures should have either recognizable structure or strong stationarity (typically white noise), so that the human brain can easily separate the two channels (noise and face). We test different

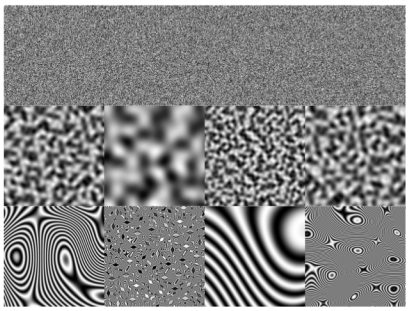

Fig. 2. Examples of noise patterns. First row: white noise, middle row: Perlin noise, last row: sine Perlin noise.

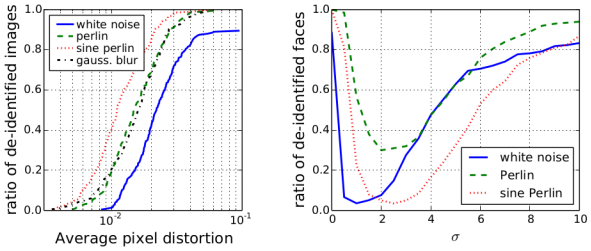


Fig. 3. *Left:* Proportion of de-identified faces as a function of the average pixel distortion. *Right:* Impact of low-pass filtering on textured forgeries.

textures and report results for three of them: white noise, Perlin noise, and what we call sine Perlin noise. Fig. 2 shows some samples. Regarding white noise images, each pixel intensity is obtained by drawing uniform integer in the range $[0, 255]$. Perlin noise is obtained with the modified algorithm of Perlin proposed in [17]. This noise is widely used in computer graphics as the random seed to generate a wide range of useful textures. We adapt the implementation of the `battlestar-tux` game project[1], which works by creating a fixed map *observed* at different locations and scales. There are 3 parameters $(x, y, s)$ which are the coordinates of the center of the texture patch and the scale of observation. The last family is a Perlin texture modulated by the sine function. It has the 3 same parameters plus the frequency of the sine. The succession of dark and light lines corresponds to the level sets of the original Perlin texture in Fig. 2.

For each image to be de-identified, we draw $N = 50$ texture images of the chosen family sharing the same dimensions as LFW images with randomly chosen sets of parameters. The line search (line 10 of Alg. 1) tests 7 values of $\eta$ anytime requiring a surjection onto the boundary approximated by a bisection with 10 iterations. In total, 3500 calls to the face verification system is needed to de-identify a face image.

### A. Self de-identification

This first set of experiments is a direct application of the method presented in Sect. III to 220 randomly selected faces from LFW. In this particular case, the reference image $\boldsymbol{x}_o$ and the starting image $\boldsymbol{x}_f$ are the same, *i.e.* $\boldsymbol{x}_o = \boldsymbol{x}_f$. These experiments used the face recognition of [11], whose detection threshold is set as the optimal threshold on the view 1 of LFW. Note that, since the altered image obtained with our method

[1] `http://code.google.com/p/battlestar-tux/`

Fig. 4. Examples of de-identified images for, from top to bottom, white noise, Perline and sine Perlin textures. The bottom line shows images de-identified using gaussian blur.

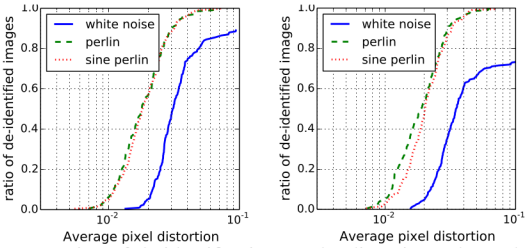


Fig. 5. Proportion of de-identification *vs.* the distortion, when robustness to low-pass filtering is enforced for $\Sigma = \{2\}$ (left) and $\Sigma = \{2, 4\}$ (right).

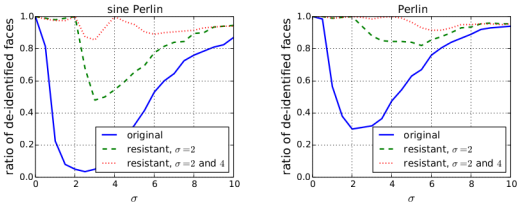


Fig. 6. Impact of low-pass filtering on robust forgeries.

are just beyond the boundary of the detection region $\mathcal{V}_{\boldsymbol{x}_o}$, slight variations could bring them back into $\mathcal{V}_{\boldsymbol{x}_0}$. To avoid this, we actually multiply the value of $\alpha(\boldsymbol{\beta})$ by a factor 1.1 before applying the final texture to the image. Fig. 3 (Left) shows the cumulative distribution of the proportion of de-identified images as a function of the average pixel distortion ($\frac{\|\boldsymbol{x}_f - \boldsymbol{y}\|^2}{255^2 P}$ with $P$ the number of pixels).

The sine Perlin textures de-identify a larger amount of images with less distortion that the other textures. Note also that the white noise textures saturate on the graph at a value lower than 1.0. This means that some images could not be de-identified using white noise. Fig. 4 displays some sample images in this context. De-identification by blurring clearly prevents human identification. This is not the case with the oracle attack using Perlin sine noise. However, the visual quality is not good. A simple trick could be to remove manually the noise except over the faces to be de-identified.

Overall, the oracle attack performs betters but achieves mitigated results when $\boldsymbol{x}_o = \boldsymbol{x}_f$. In other words, it is hopeless to de-identify an already published picture while preserving visual quality because face verification algorithms are too robust. Nevertheless, this setup is very pessimistic: users usually post new pictures on social networks.

### *B. Improving robustness to simple counter attacks*

To challenge the de-identification, we examine if the face verification system can mount simple counter attacks. Since our method introduces high frequencies, a simple low-pass filter could make the forged images recognizable again.

We test a simple gaussian filter parametrized by its standard deviation $\sigma$. The right part of Fig. 3 shows the evolution of the rate of forged images de-successfully identified as a function of the $\sigma$. The rate falls down at small $\sigma$ because the filter succeeds in partially removing the noise while preserving the face, and then rises up at bigger $\sigma$ because the filter blurs so much the image that the face can't be recognized. Perlin noise is clearly more robust to this counter attack. Our explanation is that its spectrum (density of power over frequencies) resembles more the spectrum of the face images, so that the filter has more difficulty in separating the face and the noise. for instance, a Wiener filtering would let the image untouched.

We partially achieve robustness against this counter attack by explicitly taking it into account during the oracle attack. We choose a set of $S$ scale parameters $\Sigma = \{\sigma_i\}_{i=1}^S$ for which we want to enforce de-identification, and the oracle attack considers this new verification region: $\mathcal{V}_{\boldsymbol{x}_o,\Sigma} = \mathcal{V}_{\boldsymbol{x}_o} \cup \mathcal{V}_{\boldsymbol{x}_o,\sigma_1} \cup \cdots \cup \mathcal{V}_{\boldsymbol{x}_o,\sigma_S}$ where $\mathcal{V}_{\boldsymbol{x}_o,\sigma}$ is the set of images which are identified as the person of $\boldsymbol{x}_o$ after the filtering by a Gaussian kernel of deviation $\sigma$. Since $\mathcal{V}_{\boldsymbol{x}_o} \subset \mathcal{V}_{\boldsymbol{x}_o,\Sigma}$, $\boldsymbol{y}^\star$ may get further away from $\boldsymbol{x}_f$ resulting in a greater or equal average pixel distortion, as shown in Fig. 5. When the standard deviation of the low-pass filter used at the face verification side belongs to $\Sigma$, we achieve a perfect robustness as the rate of de-identification equals 1, otherwise we drastically reduce the impact of the counter attack as shown in Fig. 6.

Fig. 7. Images de-identified using a different reference image with Perlin noise. Rows correspond to images respectively taken around 25%, 50% and 75% of the average distortion distribution.

Fig. 8. The reference and the JPEG compressed image of the same person.

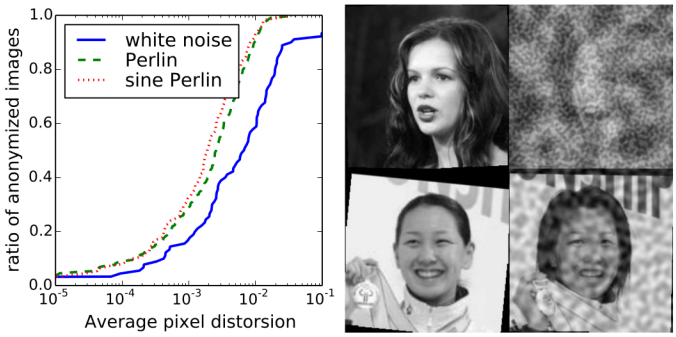


Fig. 9. *Left:* Proportion of de-identification *vs.* distortion, with low pass filtering ($\Sigma = \{2, 4\}$). *Right:* Examples of pairs where human failed.

TABLE I
CONFUSION MATRIX OF THE CLASSIFICATION OF 100 POSITIVE AND 100 NEGATIVE PAIRS BY A GROUP OF 117 PEOPLE.

| | ground + | ground − |
|---|---|---|
| human + | 93.2% | 7.2% |
| human − | 6.8% | 92.8% |

### C. De-identification of image pairs

This new set of experiments is closer to the targeted use case. The objective is now to forge an image which can't be matched with a different image of the same person: $\boldsymbol{x}_f \neq \boldsymbol{x}_o$. These experiments take the image pairs of LFW, which are correctly detected as positive by the face verification algorithm [11]. The amount of noise necessary for the de-identification is lower than in Sect. IV-A since the two face images are already different (in illumination, pose, etc.). Indeed, the noise is almost invisible for the pairs 'on the edge' of the face verification capacity (see the first row of Fig. 7). Comparatively, Fig. 8 shows that a JPEG compression not only introduces very annoying blocky artefacts, but also fails in de-identifying any image! The average pixel distortion spreads over a wider range of values (see Fig. 9 (Left)), and is globally weaker than under the $\boldsymbol{x}_o = \boldsymbol{x}_f$ setup: Half of the images are de-identified at a distortion lower than $\approx 2.10^{-3}$ whereas the median is at $\approx 2.10^{-2}$ in Fig. 5 (Right).

The other goal of these experiments is to assess whether humans still recognize people in the forged images. Pairs of images $(\boldsymbol{x}_o, \boldsymbol{y})$ are shown to humans, who are asked to evaluate if both the images are from same person or not. We display 200 random pairs of images with an equal number of positive and negative pairs of images in random order to a group of 117 people. Table I shows that our method does not alter too much the facial features important for the human brain. However, there are few pairs where our method still damages a lot the image: these are the pairs where the conditions are similar (same facial expression, same illumination, same pose *etc.*) as shown in Fig. 9 (Right).

## V. CONCLUSION AND DISCUSSION

The need for tools enabling privacy on social networks has motivated a novel approach for the de-identification of face images. Experiments show that our method achieves this goal most of the times, provided that the oracle attack uses a close enough version of the face verification system.

Our concusion holds for a given system, which is representative and state-of-the-art in this field. Will it still prevail in the future as the performances of face recognition will increase, in particular thanks to a better robustness against noise? It is well known in computer vision that there is a trade-off between robustness and discriminability. This paper illustrates this to some extend with the low-pass filtering. This counter-attack fails if the noise patterns have the same spectral power density as the faces in the images. Then, getting rid off the noise causes erasing discriminant facial traits.

## REFERENCES

[1] D. S. Bolme, J. R. Beveridge, M. Teixeira, and B. A. Draper. The csu face identification evaluation system: its purpose, features, and structure. In *Proceedings of the 3rd international conference on Computer vision systems*, ICVS'03, pages 304–313, 2003.

[2] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of Dimensionality: High-dimensional Feature and Its Efficient Compression for Face Verification. In *Computer Vision and Pattern Recognition, 2013. Proceedings CVPR'13., IEEE Computer Society Conference on*. IEEE, 2013.

[3] P. Comesaña, L. Pérez-Freire, and F. Pérez-González. The return of the sensitivity attack. In *Proceedings of the 4th international conference on Digital Watermarking*, IWDW'05, pages 260–274, Berlin, Heidelberg, 2005. Springer-Verlag.

[4] P. Comesaña, L. Pérez-Freire, and F. Pérez-González. Blind Newton Sensitivity Attack. *Information Security, IEE Proceedings*, 153(3):115–125, 2006.

[5] B. Driessen and M. Dürmuth. Achieving anonymity against major face recognition algorithms. In *Communications and Multimedia Security*, pages 18–33, 2013.

[6] F. Dufaux and T. Ebrahimi. A framework for the validation of privacy protection solutions in video surveillance. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, 2010.

[7] J. W. Earl. Tangential sensitivity analysis of watermarks using prior information. In *Proc. SPIE 6505, Security, Steganography, and Watermarking of Multimedia Contents IX*, pages 650519–650519–12, 2007.

[8] R. Gross and L. Sweeney. Towards Real-World Face De-Identification. In *First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, 2007.

[9] R. Gross, L. Sweeney, F. D. la Torre, and S. Baker. Semi-supervised learning of multi-factor models for face de-identification. In *CVPR*. IEEE Computer Society, 2008.

[10] 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.

[11] S. U. Hussain, T. Napoléon, and F. Jurie. Face recognition using local quantized patterns. In *British Machive Vision Conference*, 2012.

[12] M. Jaggi. *Sparse Convex Optimization Methods for Machine Learning*. PhD thesis, ETH Zürich, 2011.

[13] M. Jaggi. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In *Proc. of Int. Conf. on Machine Learning (ICML)*, pages 427–435, 2013.

[14] E. Learned-Miller. http://vis-www.cs.umass.edu/lfw/results.html, 2013.

[15] A. Mignon and F. Jurie. PCCA: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 2666–2672, 2012.

[16] E. M. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying face images. *Knowledge and Data Engineering, IEEE Transactions on*, 17(2):232–243, 2005.

[17] K. Perlin. Improving noise. *ACM Trans. Graph.*, 21(3):681–682, July 2002.

[18] J. P. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET Evaluation Methodology for Face-Recognition Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[19] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher Vector Faces in the Wild. In *British Machine Vision Conference*, 2013.

[20] Y. Welinder. A face tells more than a thousand posts: Developing face recognition privacy in social networks. *Harvard Journal of Law and Technology*, 26(1), 2012.