Image and Vision Computing 113 (2021) 104245

Contents lists available at ScienceDirect

Image and Vision Computing

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

A new perceptual hashing method for verification and identity classification of occluded faces

Rubel Biswas [a,b,*], Víctor González-Castro [a,b], Eduardo Fidalgo [a,b], Enrique Alegre [a,b]

[a] Department of Electrics, Systems and Automation Engineering, Universidad de León, León, Spain
[b] Researcher at INCIBE (Spanish National Cybersecurity Institute), León, Spain

ARTICLE INFO

Article history:
Received 4 March 2021
Received in revised form 20 May 2021
Accepted 13 June 2021
Available online 26 June 2021

Keywords:
Face verification
Face classification
Adversarial attack eye region occlusion
Perceptual hashing
Discrete cosine transform (DCT)
OSF-DNS

ABSTRACT

Recently, research communities on Computer Vision and biometrics have shown a lot of interest in face verification and classification methods. Fighting against Child Sexual Exploitation Material (CSEM) is one of the applications that might benefit most from these advances. In CSEM, discriminative parts of the face, i.e. mostly the eyes, are often occluded to make the victim identification more difficult. Most of the current face recognition methods are not able to handle such kind of occlusions. To overcome this problem, we present One-Shot Frequency Dominant Neighborhood Structure (OSF-DNS), a new perceptual hashing method that shows advantages on two scenarios: (a) occluded face verification, i.e., matching occluded faces with their non-occluded versions, and (b) face classification, i.e., getting the identity of an occluded face by means of a classifier trained with the non-occluded faces using the perceptual hash codes as feature vectors. We have compared the face verification performance of OSF-DNS with three perceptual hashing methods and with the features obtained from five deep learning techniques, using the occluded versions of six different facial datasets. The proposed method achieves accuracies between 69.24% and 99.46% depending on the dataset, and always higher than the compared methods. For the face classification task, we compared the performance of OSF-DNS with the features obtained by four deep learning techniques. Experimental results on LFW and CFPW datasets showed that the proposed hashing method outperformed the results obtained with deep features with an accuracy up to 89.53%.

1. Introduction

Automatic face verification, i.e., the task of matching a given face with another, and classification, i.e., obtaining the identity of a person on a digital image by means of a trained model, are areas with substantial active research in the Computer Vision community. Moreover, they have attained remarkable performance for different applications in recent years with the development of traditional methods, e.g. face detection [1], verification [2], recognition [3] or alignment [4] and recognition based on deep learning methods [5–8].

Applications such as forensics, surveillance, or offenders' or crime victims' identification require solutions to address the problems of detecting whether a face is within a database or identifying a person [5, 9–11] with good performance. Unfortunately, such performance drops when the face is occluded, especially when the eyes are covered (intentionally or not) by a mask or any object – e.g., glasses –, or because of adversarial attacks [12]. An adversarial attack consists of modifying an image with the intention of inserting alterations/perturbations on the image in order to deceive a classifier [12]. In this paper, we call "Adversarial Eye Region Occlusion" (AERO) to the attack in which the eyes of a person in an image have intentionally been occluded. The AERO attack is one of the most grievous attacks when dealing with face verification or classification tasks. Fig. 1 depicts some examples of AERO attacks.

Some approaches to handle the occlusion challenge in the face recognition task have been presented. There are frameworks based on auto-encoders [13,14], that remove the occluded parts of the face prior to the recognition. Other approaches have proposed local feature learning-based methods, like constraints-based dictionary learning [15] and reconstruction-based methods [16]. Approaches based on local feature learning make local features mutually independent where the occluded area may not affect the rest of the parts. By concerning the unbalanced difficulty between the negative and positive samples, some approaches have been presented, such as sparse coding with manifold learning [17], or kernel prototype similarities [18]. One of the main limitations of this kind of approach is that the augmented examples are extremely correlated to the original ones. Additionally, most of the mentioned methods require training data of both natural and occluded faces of a person to carry out occluded face verification.

* Corresponding author at: Department of Electrics, Systems and Automation Engineering, Universidad de León, León, Spain.
E-mail addresses: rbis@unileon.es (R. Biswas), victor.gonzalez@unileon.es (V. González-Castro), eduardo.fidalgo@unileon.es (E. Fidalgo), enrique.alegre@unileon.es (E. Alegre).

https://doi.org/10.1016/j.imavis.2021.104245
0262-8856/

Fig. 1. Examples of AERO attacked versions (bottom) of some faces (top).

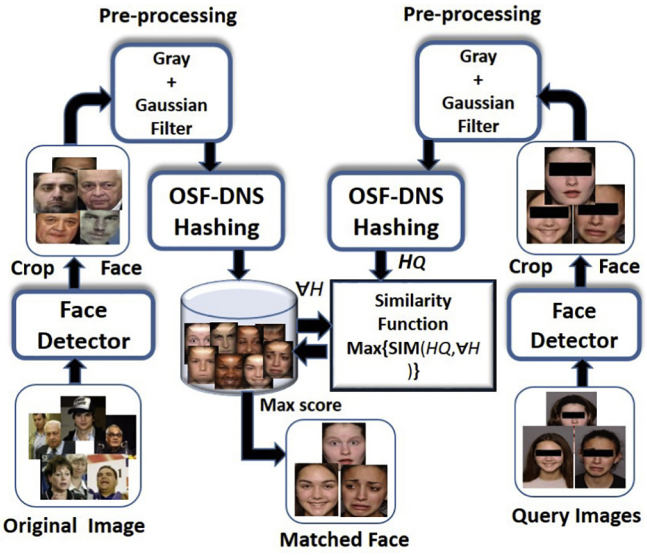

Fig. 2. Face verification process against the AERO attack. First, the occluded faces within an input image are detected and cropped using a face detection algorithm. Then, these faces are preprocessed and the OSF-DNS hash codes are extracted from them. Finally, these codes (H_Q) are compared with the codes previously extracted from the original faces ($\forall H$), which are stored into the system storage by means of a similarity function (**SIM**), to find the most similar non-occluded face.

A fast detection of Child Sexual Exploitation Material (CSEM) is essential, either to prevent its sharing as soon as possible or to make a legal case against presumed offenders. However, manual detection of such material is time consuming and also disturbing for officers that have to fulfill that task. Therefore, automatic CSEM detection [19] would be a great support for Law Enforcement Agencies (LEAs). Automatic face verification or classification may be helpful in this kind of cases. Verifying that the face of a person that appears on an image is already on LEAs' databases of Child Sexual Abuse (CSA) cases or identifying a person on an image may lead to interesting insights for LEAs, from victim identification to establishing links between different CSA cases. Unfortunately, sometimes AERO attacks are applied to CSA materials by offenders, thus making automatic face verification and classification more difficult, which makes it a challenge for LEAs in cases involving CSA.

Consequently, an automatic face verification or classification method robust against adversarial attacks, especially the AERO attack, would be very convenient [20]. We believe that perceptual hashing [21] may be a suitable approach to achieve face verification and that the hashing features may improve the occluded face classification performance. Perceptual hashing represents an image by means of its content as a sequence of binary or real numbers, called hash code. It has been utilized in real applications such as multimedia forensics [22], copy detection [23] or retrieval [24]. This technique generates the same or very similar hash codes even if any content-preserving operation has previously been applied to them, e.g. scaling, noise addition, compression, or geometric transformations. On the other hand, visually different images should be represented by different hash codes.

In this paper, we present a new perceptual hashing method called One-Shot Frequency-Dominant Neighborhood Structure (OSF-DNS), on the basis of our previous work [25]. We have applied it for (1) robust face verification and (2) face identity classification, in both cases dealing with AERO attacked faces. They are depicted in Figs. 2 and 3, respectively. To carry out the face identity classification robust to the AERO attack, the OSF-DNS hash codes of the original, i.e. non-occluded, faces were used to train a Support Vector Machine (SVM) classifier to further predict the identities of occluded faces, also described by means of OSF-DNS. Both approaches can be applied to enhance the ability of forensic tools to verify or classify whether an AERO attacked face in CSEM is the same as a face found in other images, e.g. seized materials in previous CSA cases. As a result, the face verification method or face classification scheme might be a good assistance to LEAs for CSA material detection.

The contributions of this paper are summarized below:

- We introduce a new perceptual hashing method, called OSF-DNS. It shows good performance for the AERO attacked face verification task, compared with other perceptual hashing and deep feature-based methods.
- We also utilize the proposed OSF-DNS hashing features to classify an occluded face, i.e., get the identity of an AERO attacked face. Compared with the features obtained by other perceptual hashing methods and deep learning methods, OSF-DNS outperforms them in this task.
- We present a framework for the problem of victim identification into CSEM, especially when those victims have their eyes manually covered by offenders.
- We create the AERO attacked version of six state-of-the-art datasets. To compare the face verification and classification performance. We also compare the face verification and classification performance of OSF-DNS over these datasets with other three hashing methods, and with five state-of-the-art CNNs focused on face recognition.

The rest of this paper is structured as follows. Section 2, presents a brief revision of the state-of-the-art in face recognition and perceptual hashing. The AERO attack face database creation process is presented in Section 3. Section 4 describes the proposed OSF-DNS method. The process by which the proposed method is used for face verification and classification against AERO is described in Section 5. The experimental results and performance comparisons are presented in Section 6 and, finally, the conclusion and future work are discussed in Section 7.

2. Related work

2.1. Face recognition

From the last few years, significant development has been noticed on face recognition research [5,7,11,26], mainly due to the availability of massive data and GPUs to train Deep Learning models. Some existing approaches [27,28] have also focused on face recognition with expression, pose, aging, disguise, or illumination changes. One of the most challenging situations for face recognition is when the face is partially

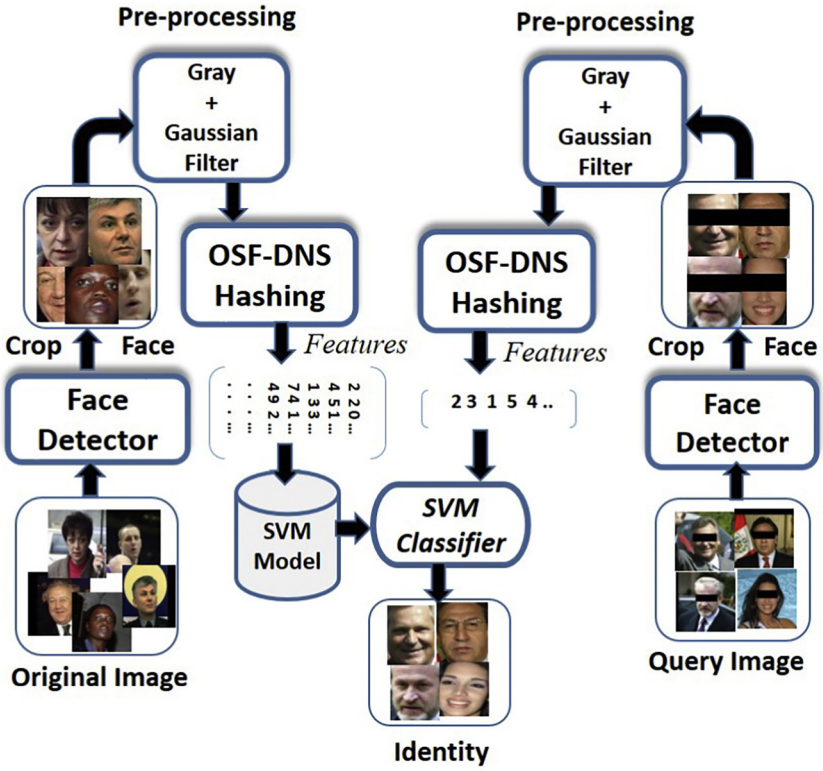

Fig. 3. Face classification process against AERO using OSF-DNS features: First, the non-occluded faces of a dataset are cropped and then their OSF-DNS hash features are extracted. These features are then used to train a SVM classifier. Afterwards, given an AERO-attacked face, it is detected and cropped to compute its OSF-DNS hash feature vector, which is finally classified by the trained SVM model to get its identity.

occluded, either naturally – e.g. by a hand or an object – or artificially, i.e., when the image is edited to cover the eyes or other part of the face.

To the best of our knowledge, few works deal with face verification or classification robust to occlusions. For instance, Morelli Andrés et al. [29] employed compressed sensing to detect and remove the occluded part from the face image. More specifically, they used local features to generate a new, non-occluded, image that is similar to the one they wanted to recognize. Then a query image was subtracted from this image to detect the occlusion area through a threshold. In the end, only the non-occluded pixels were used to recognize the identity. Recently, Koç M. [30] introduced a method based on the same idea. Specifically, they used three coefficients to detect the occluded partitions, i.e., entropy, correlation, and root-mean-square error of the partition. Later, they combined the scores given by classifier for a class at each of the non-occluded regions using the sum, product, and majority vote rules to get the identity. Dagnes et al. [31] proposed a method for 3D face recognition robust to the eye and mouth occlusions due to hands. Such occlusions were detected and removed by exploiting the 3D geometry, i.e., by considering their effects on the 3D points. Finally, the non-occluded symmetrical regions have been used to restore the missing facial information for recognizing the face. Domingo et al. [32] presented a method for unconstrained face recognition robust to ambient lighting, pose, expression, occlusion, face size and distance from the camera based on dictionary learning and Sparse Representation Classification. Wu et al. [33] proposed a method, called occlusion pattern-based sparse representation classification (OPSRC), to learn the occlusion pattern from the query data. Mustafa et al. presented an occluded face recognition framework [34] based on the two-dimensional multi-color fusion (2D-MCF) representation and the Partitioned-sparse sensing recognition (P-SRC) classifier.

Local feature learning is another approach to deal with occlusion. In this idea, features are extracted from local areas of the face image, and they are used for the recognition through a locally matching strategy. Based on this concept, Liao et al. [35] presented the Multi-Keypoint Descriptors (MKD) to represent the alignment-free face where the actual content of the image determines the size of the descriptor.

To recognize the partially occluded face, Duan et al. [36] proposed a scheme based on topology-preserving graph matching to estimate more accurate and robust topological information. It has estimated a non-rigid transformation encoding the second-order geometric structure of the graph.

Non-negative matrix factorization (NMF)-based learning provides an effective way for face recognition robust against occlusions. An example is the dictionary learning method proposed by Ou et al. [37]. They made low-dimensional representations of samples from the same class to be as close as possible to enhance the discriminant ability of the dictionary.

A LSTM-autoencoders model was introduced by Zhao et al. [38] that consists of a multi-scale spatial LSTM encoder to generate an occlusion-robust representation of the face and a dual-channel LSTM decoder to recurrently remove the occlusion in the image space.

Xu et al. [39] presented a framework based on pose-guided spatial attention (PGSA) and activation-based attention (AA), called dual-attention re-identification (DAReID), to obtain more robust features for re-identifying the occluded person. They mainly used PGSA and AA to obtain (i) robust local features of the visible and occluded regions of a person and (ii) global features of the visual activation levels of different regions, respectively. Then, combined them with human pose information to define weighted local distances (WLD) to learn new and more discriminative local features for the re-identification task.

Additionally, several recent approaches have addressed the occlusion problem by employing low-rank representations [40], hierarchical sparse and low-rank representations [41], a discriminative multi-scale sparse coding (DMSC) [42] and fuzzy max-pooling to solve double-occlusion problem [43].

2.2. Perceptual hashing

Perceptual image hashing is a specific research field which plays an important role in various Computer Vision applications such as image authentication, image description, or image copy detection [23,44]. Ideally, it should be robust against certain types of attacks, e.g., noise addition, contrast adjustment, digital watermarking, scaling, or rotation [45, 46]. Many perceptual hashing methods employ the Discrete Cosine Transform (DCT), the Discrete Wavelet Transform (DWT), or the Discrete Fourier Transform (DFT).

Moreover, many perceptual image hashing methods have been introduced in the field of multimedia security [25,45], some of them showing better performance than image classification [47,48]. Qin et al. [45] proposed a hashing method that considered the texture and color features of the image through Weber local binary pattern and color angular pattern for representing perceptual image content. Davarzani et al. [49] used the center-symmetrical local binary pattern (CSLBP) to generate the image hash, proving to be useful for tampering detection. Tang et al. [50] presented a scheme to build an image hash by employing the histogram of color vector angles.

Recently, Yuan and Zhao [51] introduced a hash algorithm that combined three-dimensional global features and local energy features. Firstly, they compressed the input image using Singular Value decomposition (SVD) and then extracted statistical features – i.e., global features – at the three-dimensional visual angle and the energy variation features – i.e., local features – in the four directions. Finally, these global and local features were combined and scrambled to obtain the final hash.

Fang et al. [52] described a person's image by exploiting the perceptual hash-based features. In their work, low-level color and gradient features were extracted to create a hash feature map. Later, the histogram, mean vector, and the co-occurrence matrix features were computed from the center area of the map to re-identify a person. A perceptual image hashing scheme [20] was presented by Biswas et al. for occluded face identification purposes. This method combined the frequency coefficients and statistical image information to construct the face hash.

As we have stated, a large portion of the occluded face recognition methods [37,43] include a model that has to be trained with facial images, including occluded ones. To avoid building a model for occluded face verification, we have designed a perceptual hashing method (named OSF-DNS) robust to partial occlusion in the eye region. In addition, we also present a classification pipeline to get the identity of an occluded face with no need of training the model with occluded faces. This method uses the OSF-DNS codes as face features.

3. Adversarial eye region occlusion attack face database creation

There are a lot of datasets for face recognition purposes (some of them mentioned in Section 6.1), but there are no occluded face datasets. Therefore, in order to evaluate the face verification or classification performance of our proposed hashing method and state-of-the-art methods and their robustness to occlusions, we have had to create an AERO attack face dataset. In this Section, we present an automatic procedure to carry out this task on the basis of existing face datasets.

To build the AERO attack version[1] of the faces in an image, we first applied the Multi-task Cascaded Convolutional Networks (MTCNN) face detector [53] to locate them, except for those of the LFW dataset, in which we used the bounding boxes provided in the Ground Truth. Afterward, using the eye landmarks detected by MTCNN, we selected the most external points of the left and right eyes, denoted by P_1 and P_2, respectively. Next, we defined two ratios, R_1 and R_2, over P_1 or P_2, respectively. R_1 defines the occlusion mask height, leaving points P_1 or P_2 as the center of the mask on the left and right borders, respectively. R_2 is

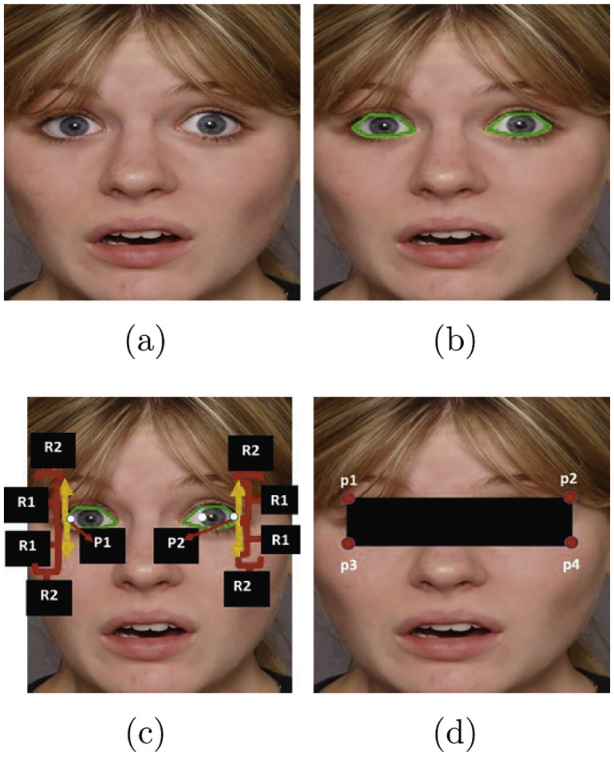

Fig. 4. Stages of the automatic eye region occlusion. Given a face image (a), first the eye landmarks are found by means of MTCNN (b). Afterwards, the most external points of the eyes and the ratios (c) are used to calculate the corners of the occlusion mask (d).

taken from the point of P_1 to the left and from the point P_2 to the right to define the mask width.

In this paper, we have empirically set the value of $R_1 = 20$ and $R_2 = 5$ during the preparation of the occluded face dataset. Finally, we obtained the top-left, top-right, bottom-left, and bottom-right points, named p_1, p_2, p_3 and p_4, respectively, of the eye occlusion rectangle. Let (P_{1_x}, P_{1_y}) and (P_{2_x}, P_{2_y}) be the coordinates of P_1 and P_2, respectively. Then, the corners of the occlusion rectangle are defined as:

$$p1 = \left(P_{1_x} - R_1, P_{1_y} - R_2\right) \tag{1}$$

$$p2 = \left(P_{2_x} - R_1, P_{2_y} + R_2\right) \tag{2}$$

$$p3 = \left(P_{2_x} + R_1, P_{2_y} - R_2\right) \tag{3}$$

$$p4 = \left(P_{1_x} + R_1, P_{1_y} + R_2\right) \tag{4}$$

Fig. 4 depicts the occluded face creation process.

We want to remark that during the occlusion process, several faces were not detected by MTCNN, e.g., non-frontal faces in the CFPW dataset. Once the face detection process finished, we visually inspected the results and manually occluded those for which the face detection failed.

Note that we have designed the automatic procedure to generate occlusion over the eye region so that the generated mask looks like the ones which can be found in real cases in CSEM.

Fig. 5 presents some examples of the AERO attack faces from six different datasets, i.e. LFW [54], CUHK[2], MEDS-II [55], CFPW [56], VGGFace2[3] and NIMH-ChEFS [57].

[1] http://gvis.unileon.es/dataset/occluded_face_generator/

[2] http://mmlab.ie.cuhk.edu.hk/archive/facesketch.html

[3] http://www.robots.ox.ac.uk/vgg/data/vgg_face2/

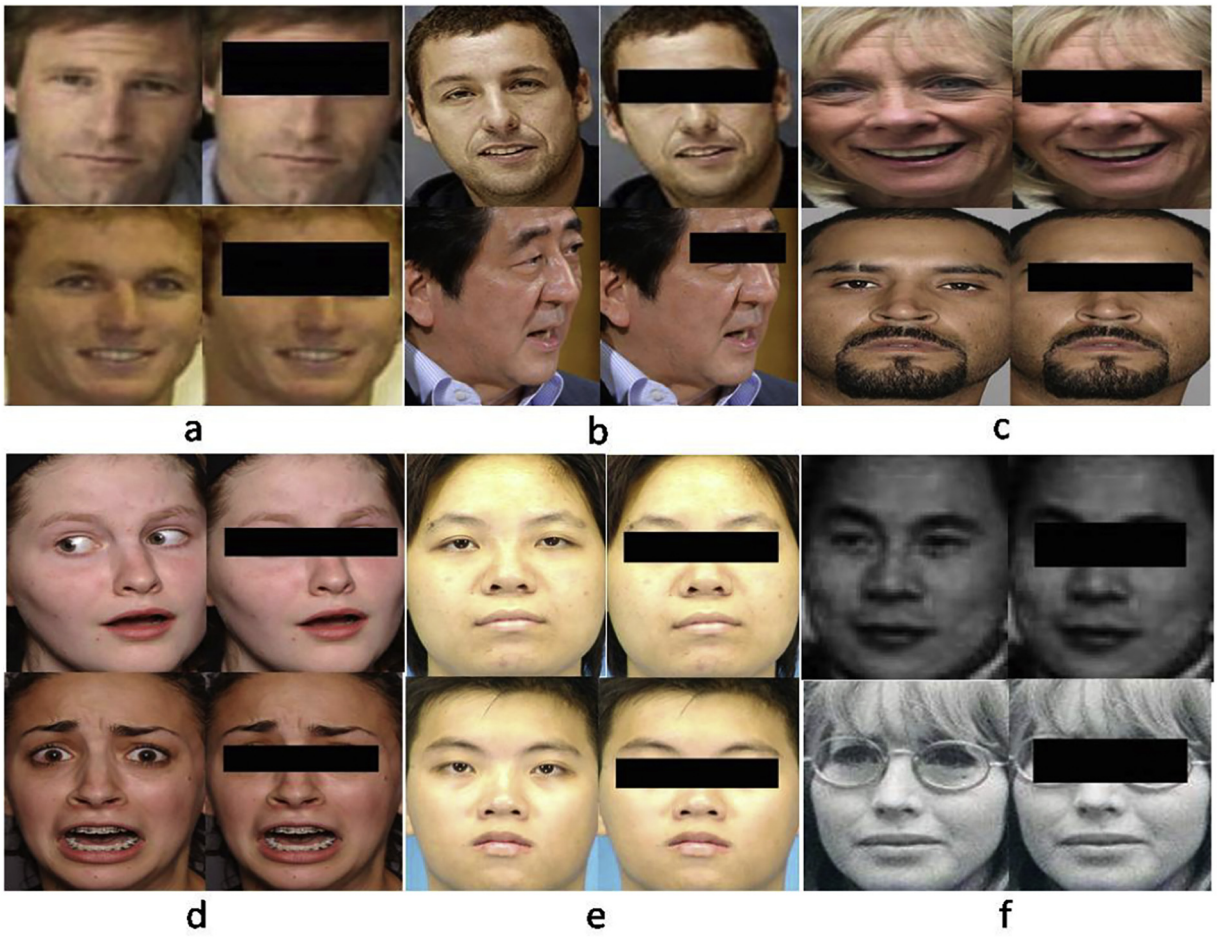

Fig. 5. Examples of the AERO attack face images from six different datasets: LFW (a), CFPW (b), MEDS-II (c), NIMH-ChEFS (d), CUHK (e) and VGGFace2 (f).

We also analyzed the resolutions of the images of these datasets and computed the average and standard deviation (std) Euclidean distance between the eyes of the faces in these datasets, which are reported in Table 1. With respect to the average spatial resolution of the images in the datasets, we noticed that LFW contains the smallest images (i.e., 250 × 250), whereas NIMH-ChEFS includes the biggest ones, i.e., 1960 × 3008. Subsequently, it can be observed that the average and std. Euclidean distance between the eyes in LFW are lower (41.41,2.60). This low std. is compatible with the fact that the LFW dataset contains face images with fewer variations in terms of pose and illuminations. On the contrary, the average distance and std. between eyes of the faces from the NIMH-ChEFS dataset are higher, (524.83, 47.23), than those from the rest of the datasets, i.e., LFW, CFPW, MEDS-II, CUHK, and VGGFace2.

4. One-shot frequency-dominant neighborhood structure (OSF-DNS)

In this Section, we introduce the proposed perceptual hashing scheme, which we named One-Shot Frequency-Dominant Neighborhood Structure (OSF-DNS). Fig. 6 depicts the main steps of the OSF-DNS method, which are explained in detail in the next subsections: image pre-processing, feature extraction and hash generation.

4.1. Image pre-processing

The first step after detecting a face is to crop it. The MTCNN method returns, for each detected face, the coordinates of the bounding box that contains it (i.e., (x, y, *width*, *height*), where, (x,y) is the bottom left corner and (*width*, *height*) are the width and the height, respectively). We used these coordinates to crop the detected faces automatically. Then, the cropped face image I is resized to 120 × 120 pixels by means of the *resize* function of the Python library *scikit-image*, which uses a bi-linear interpolation. Then, it is converted to greyscale and smoothed using a Gaussian low-pass filter G (Eq. (5)).

$$G(m,n) = \frac{g(m,n)}{\sum_{m=1}^{M}\sum_{n=1}^{N} g(m,n)} \tag{5}$$

where:

$$g(m,n) = \frac{e^{-(m^2+n^2)}}{2\sigma^2} \tag{6}$$

where σ is the standard deviation of the Gaussian convolution mask. Let I_G be the resulting image. The Gaussian low pass filter smooths the image in both homogeneous and heterogeneous areas of the image, i.e., edges and textures. This helps to eliminate some noise in the image as well as to preserve image quality without losing its principal features.

Table 1
Average Euclidean Distance (ED) and Standard Deviation (Std) of detected face eyes distance for six datasets with average input image resolutions.

Dataset	Avg. resolution	Avg. ED	Std ED
LFW	250 × 250	41.41	2.60
CFPW	441 × 370	84.18	45.31
MEDS-II	480 × 600	108.15	42.71
NIMH-ChEFS	1960 × 3008	524.83	47.23
CUHK	1024 × 768	141.39	6.45
VGGFace2	388 × 360	48.30	29.91

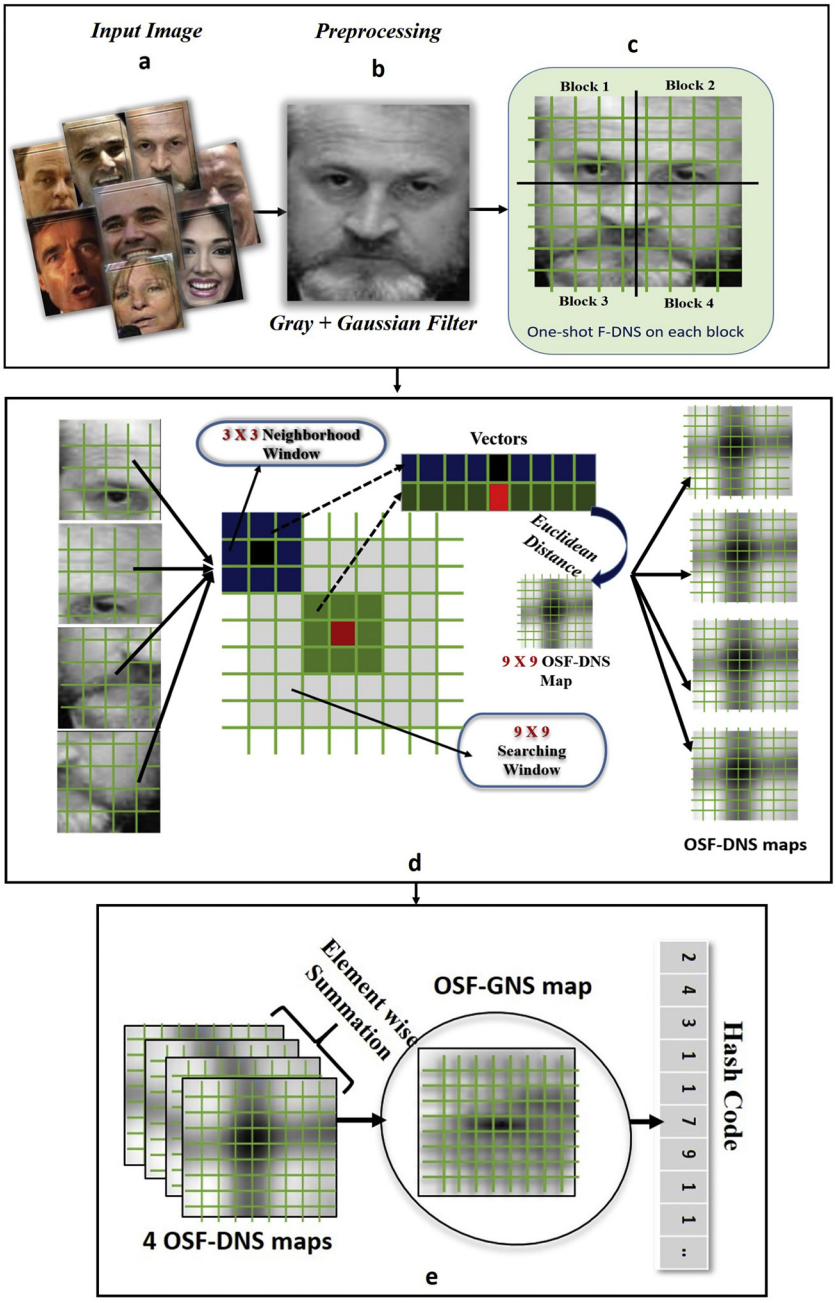

Fig. 6. Overview of OSF-DNS perceptual hashing computation: First, the input image (a) is pre-processed (b). Next, the pre-processed image is divided into four equal blocks (c). From each block, the DCT is applied and a OSF-DNS map is computed (d). Finally, these four maps are added to get a OSF-GNS map and, subsequently, the hash code (e).

4.2. Feature extraction

After preprocessing the face image, features are extracted from it using the Discrete Cosine Transform (DCT) and the Dominant Neighborhood Structure (DNS), proposed by Khellah [58].

First, I_G is divided into four non-overlapped blocks of 60×60 pixels, i.e., I_{B_1}, I_{B_2}, I_{B_3} and I_{B_4}. Such division helps to minimize the negative influence of the occlusion in the image. Then, the DCT is applied to each block to obtain four frequency blocks, i.e., I_{DCT_1}, I_{DCT_2}, I_{DCT_3} and I_{DCT_4}.

Let f be an image of size $M \times N$[4]. The DCT of an image f, i.e. F, is represented by Eq. (7).

$$F(i,j) = \alpha(i)\alpha(j) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cos\left(\frac{(2x+1)i\pi}{2M}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right) \quad (7)$$

where $f(x,y)$ is the pixel value at the coordinate (x,y) of the image f. $F(i,j)$ represents the DCT of f, where i and j are the vertical and horizontal frequencies, respectively. In addition, $\alpha(i) = \sqrt{\frac{1}{M}}$ and $\alpha(j) = \sqrt{\frac{1}{N}}$ for $i, j = 0$, whereas $\alpha(i) = \sqrt{\frac{2}{M}}$, $\alpha(j) = \sqrt{\frac{2}{N}}$ for $i, j \neq 0$.

The DCT of an image is represented by a matrix of real numbers that contains a DC component and several AC components. The former refers to the zero-frequency component, whereas the AC components refer to higher vertical and horizontal spatial frequencies, thus capturing the frequency structure of the image. Additionally, the energy in the DCT

[4] In this work, the input face and, subsequently, the blocks are squares (i.e., 60×60). Therefore, $N = M$.

coefficients can be used as a measure of the roughness of the image. Therefore, computing local fluctuations in the magnitude of such neighbor coefficients (e.g., taking into account energy blocks) may provide information about the uniformity or coarseness of the image. Moreover, it can pack most of the information of the image in a few coefficients, and DCT-based image coders perform very well at moderate bit rates, high compression ratios, and high degradation in the image quality. Therefore, in this work, we utilize the DCT coefficients to obtain the local statistical dependencies between frequency information through OSF-DNS and OSF-GNS, capturing the complete energy distribution of the image in the frequency domain.

4.2.1. One-shot frequency-dominant neighborhood structure (OSF-DNS)

Afterwards, the Dominant Neighborhood Structure (DNS) [58] is applied to each frequency block to extract features from its texture by exploiting the energy similarity between coefficients. The DNS, D, represents the degree of energy similarity between a pixel in the image, called *central pixel*, and its neighbor pixels. Let p be the *central pixel*. To calculate its DNS D_p, we measure the intensity similarity between p and each pixel p', placed within a $S_w \times S_w$ neighborhood around p, called *searching window*. This is done by calculating the Euclidean distance between the intensities of the flattened $N_w \times N_w$ matrices, called *neighborhood windows*, around p and each p'. Let the coordinates of p' within the *searching window* be (x,y). Then, the similarity between p and p' is placed in the position (x,y) of the DNS, i.e. $D_p(x,y)$.

In this work, the size of the neighborhood windows is 3×3 pixels (i.e. $N_w = 3$), and the size of the searching window is 9×9 pixels (i.e. $S_w = 9$). Therefore, the size of the resulting DNS map is 9×9 pixels. In our previous work [25] we extracted several DNS maps from the whole image, leaving horizontal and vertical gaps between the central pixels of the searching windows. In this work, we have used a single searching window per block, which covers its 9×9 top-left DCT coefficients. As a result, from each DCT-transformed block of the image, a single DNS map is extracted, i.e. D_{DCT_j}. For this reason, we have named our proposed method OSF-DNS. Finally, four 9×9 OSF-DNS maps, i.e., D_{DCT_1}, D_{DCT_2}, D_{DCT_3} and D_{DCT_4}, are extracted from the input face image.

Fig. 6 represents the OSF-DNS calculation steps of a face image. To generate a distinctive neighborhood structure of the image, it is important to generate a OSF-DNS map from each of the four frequency blocks, covering the whole texture energy distribution of the image.

4.3. Hash generation

In this stage, the global texture energy features are computed by summing up the four extracted 9×9 OSF-DNS maps (see Eq. (8)). As a result, we obtained a 9×9 global feature map, named One-shot Frequency Global Neighborhood Structure (OSF-GNS). This is depicted in Fig. 6.

$$\mathrm{OSF{-}GNS} = \sum_{m \in A} D_{DCT_m} \tag{8}$$

where D_{DCT_m} is the m-th OSF-DNS map and A is the set that contains the four OSF-DNS maps extracted from an image. The OSF-GNS map approximately represents the inherent global texture energy homogeneity.

The final image hash is composed by the coefficients of the 9×9 OSF-GNS map except for those in the first row and column. Discarding these, we avoid including the DC components of the DCT of each block, which were considered in the calculation of the OSF-DNS maps. Such components might capture completely flat image information, e.g. solid colors, and, thus, have significantly different values with respect to the other components. Therefore, they would throw off the average so, they are removed. As a result, at the end of the process, the OSF-DNS hash code of each image is composed of 64 real values. We want to remark that we carried out experiments using all the 91 values of the OSF-GNS map and the results were worse than the ones obtained with hash codes of 64 elements.

Note that we have considered using bigger searching windows, e.g., 11×11 or 21×21, but we obtained longer hash codes that increased the computational time needed for carrying out the similarity comparisons between hashes. OSF-DNS required 0.0273 s when the searching window size was 9×9. On the contrary, when the searching window size was 21×21, OSF-DNS required 0.0987 s. In addition, we conducted experiments using bigger (i.e. 5×5) neighborhood windows and the results did not change significantly with respect to the chosen size, i.e. 3×3.

5. Face verification and classification against AERO using OSF-DNS

5.1. Face verification

Face verification robust to the AERO attack is the task of matching an input face image whose eye region is occluded with the corresponding non-occluded face image. In this work, we carry out this task using the proposed OSF-DNS in three stages. First, we detect the original, i.e. non-occluded, faces in the image datasets using a face detector, e.g. MTCNN [53], and we compute and store their OSF-DNS hash codes.

Afterwards, in the phase of AERO attacked face verification, the occluded faces are detected and cropped either manually or automatically, e.g., by means of an effective occluded face detection approach such as Faces Attention Network (FAN) [59]. Then, the hash code of each occluded face is extracted by means of OSF-DNS.

Finally, the extracted hash of each occluded face is compared against the stored hash codes of the non-occluded faces using the correlation coefficient similarity function (Eq. (9)). The occluded face is matched (i.e., verified) with the non-occluded face for which this similarity score is maximum. These steps are depicted in Fig. 2.

5.2. Face classification

Face classification robust to the AERO attack is carried out in three steps. First, the OSF-DNS hash codes of the non-occluded faces are extracted as it was done in the face verification task (see Section 5.1). Secondly, the hash codes of the non-occluded faces are used to train an SVM classifier.

Finally, given an occluded face, its OSF-DNS hash code is extracted as it was done in the face verification task Section 5.1) and then, it is used as a feature vector that will be classified by the trained SVM model to get the identity of the AERO-attacked face. This process is represented in Fig. 3.

6. Experimental results and performance analysis

6.1. Experimental setup

6.1.1. Face verification experiments

In this work, we have compared the performance for face verification against the AERO attack of the OSF-DNS hash with three perceptual hashing methods, i.e., Ring Partition and Invariant Vector Distance (RP-IVD) [60], Selective Sampling for Salient Structure Features (SS-S-SF) [46] and pHash [61]. In RP-IVD, we have set the number of rings to 40, the number of keys to 100, and we normalized the image size to 256×256 as it was indicated by Tang et al. [60]. For Salient Structure Features (SS-S-SF) [46], we converted the input RGB image into the CIE L*a*b* color space. In addition, we have also compared the performance of OSF-DNS for face verification against the AERO attack with the embeddings obtained by five deep learning approaches, which have been considered as hash codes in this experiment. The compared approaches have been: FaceNet [11], OpenFace [7], VGG16 [62], ArcFace [5] and the face recognition method implemented on the DLib library [63]. We have used the pre-trained models, which were trained with non-occluded

faces, simulating the real case in which the LEAs would train their models with non-occluded faces. In the case of VGG16, we excluded its fully connected layers to attain the features of a face image. In the case of Dlib face recognition [63], we used the Python Dlib module[5] to extract the embeddings.

We have used six state-of-the-art face datasets with different expressions and poses. We also created their AERO versions by occluding the eye region of their faces, as we have explained in Section 3.

1. Labeled Faces in the Wild (LFW) [54], which comprises 13,233 faces, mainly in frontal view, with 5749 identities.
2. Celebrities in Frontal-Profile in the Wild (CFPW) [56], which includes 500 subjects with 10 frontal and 4 profile images each.
3. Child Emotional Faces Picture Set (NIMH-ChEFS) [57], with 482 pictures of fearful, angry, happy, sad and neutral child faces with two gaze conditions (direct and averted).
4. Chinese University of Hong Kong (CUHK) student database,[6] with 188 face images.
5. Multiple Encounter Dataset (MEDS-II) [55], comprising a total of 1309 images of 518 subjects.
6. VGGFace2[7], from which we have only used the test dataset, which contains 169,396 images with 500 identities.

We have used Python 3 to calculate and compare the face hashes with pHash, OSF-DNS, as well as to extract the CNN-based features from the images. MATLAB was used for extracting the hashes using the RP-IVD and SS-Salient-SF methods.

To compute the similarity between hashes, we have employed the correlation coefficient similarity function, as it has been used in [46]. Let $H_1 = [h_1^{(1)}, h_2^{(1)}, \ldots, h_l^{(1)}]$ and $H_2 = [h_1^{(2)}, h_2^{(2)}, \ldots, h_l^{(2)}]$ be two face image hashes where l is the hash length. Thus, the correlation coefficient of H_1 and H_2 is calculated by means of Eq. (9):

$$S(H_1, H_2) = \frac{\sum_{i=1}^{l}\left(h_i^{(1)} - \mu_1\right)\cdot\left(h_i^{(2)} - \mu_2\right)}{\sqrt{\sum_{i=1}^{l}\left(h_i^{(1)} - \mu_1\right)^2}\cdot\sqrt{\sum_{i=1}^{l}\left(h_i^{(2)} - \mu_2\right)^2} + \xi} \tag{9}$$

where μ_1 and μ_2 are the mean values of H_1 and H_2, respectively, and ξ a small constant to avoid division by zero. The range of the correlation coefficient S is $[-1, 1]$, and the higher S, the more similar two face images are. This means that two faces can be considered as similar if the correlation coefficient score of their hashes is higher than a certain threshold T. Otherwise, faces are considered as different persons. In this work, we set $T = 0.98$ empirically.

Since the correlation coefficient is calculated using hash codes composed by real values, in the case of pHash we worked with the values obtained before binarizing the code, instead of the binary hash code yielded by the original method [61].

6.1.2. Face classification experiments

Face classification against the AERO attack experiment has been carried out by training an SVM classifier with hash codes of the natural faces used as feature vectors, being the classes the identities to which the AERO attack faces belong. We have compared the proposed OSF-DNS with the features obtained by the other perceptual hashing and the deep learning-based approaches assessed for the face verification experiments, mentioned in Section 6.1.1. In this experiment, we have also assessed the performance of the SVM with five different kernel functions: Linear, RBF, and Polynomial with degrees 2, 4, and 6 (named Poly-2, Poly-4, and Poly-6, respectively, hereafter). In addition, we have considered two labeled datasets, i.e., LFW and CFPW.

[5] https://github.com/ageitgey/face_recognition
[6] http://mmlab.ie.cuhk.edu.hk/archive/facesketch.html
[7] http://www.robots.ox.ac.uk/vgg/data/vgg_face2/

6.2. Result analysis

6.2.1. Face verification

In face verification against the AERO attack, we repeated the processstated in Section 5 for all the images from the dataset and finally computed the verification accuracy, i.e., the % of faces that matched. The accuracies obtained with OSF-DNS on the six datasets, as well as with the compared perceptual hashing, and the deep-learning methods mentioned in Section 6.1.1.

As it is stated in Table 2, OSF-DNS perceptual hashing method outperforms the other perceptual hashing methods and the five deep feature-based approaches in all datasets. In contrast, ArcFace yields the lowest accuracy in all datasets. Moreover, pHash is the hashing method that performs better (besides OSF-DNS). It is also remarkable that the assessed perceptual hashing methods achieve better results than the CNN-based methods most times, even though such methods were specifically designed for face recognition or image classification.

It can be observed that the performance of pHash is close to that achieved by OSF-DNS, specially in the datasets MEDS-II, NIMH-ChEFS and CUHK. It may be because the hash of an input face is also computed from the DCT transformed image. Mainly, pHash exploits the DCT coefficients to attain the global description and average illumination of a face image, including the attacked region. In OSF-DNS, the input face image is divided into four frequency blocks and the DCT is applied to each of them before computing the hash. Indeed, the attacked region over the face image is distributed into each of the four blocks and, therefore, measuring the fluctuations in the magnitude of local coefficients may give information with less noise about the uniformity of the block. Another reason can be the tiny number of images that may increase the probability to retrieve the target face.

In addition, the accuracy of our proposed scheme is comparatively lower in the case of CFPW and VGGFace2 datasets. The reason may be that both datasets contain faces with different poses and illuminations. Subsequently, features of some faces, e.g., non-frontal faces in CFPW, are similar to each other and, therefore, this makes the verification to be inaccurate.

It is also observed in Fig. 7 that the verification accuracies of all deep feature-based methods are generally poorer than the perceptual hashing methods, especially in the case of OSF-DNS, for all the datasets. The reason may be that occlusion leads to a distortion of the face embeddings obtained by the convolutional base of the networks, which makes discrimination more difficult. Mainly, ArcFace attains much worse performance than the rest of the CNN-based approaches. In ArcFace, a more reliable method to increase the feature distances is applied [5]: an arc-cosine function is applied in the angular domain so that the decision boundaries between features corresponding to different classes are more distant from each other. During the experiment, we mainly extracted the embedding features of a face and its AERO attack version through the pre-trained ArcFace model. Though both faces are the same except the occluded region (i.e. they belong to the same class), the method provides very different embeddings for them. The reason may be that in ArcFace, the embeddings of the faces are distributed around each feature center toward the hyper-sphere and it uses an additive angular margin penalty between feature and ground truth weight to concurrently enhance the intra-class compactness and inter-class discrepancy. In this work, however, we use the pre-trained model, which is trained with non-occluded faces, to extract the embeddings of an occluded face. Therefore, the attacked face features for the same identity are distorted, thus affecting the accuracy.

We have carried out an additional experiment to evaluate the face verification performance of OSF-DNS and the other assessed perceptual hashing methods (see Section 6.1.1) when the faces are resized to resolutions higher or lower than 120×120. More specifically, we have resized the faces detected with MTCNN to 90×90 and 180×180 before computing their hash codes. Later, we carried out the same face

Table 2
Face verification accuracy (in %) against the AERO attack OSF-DNS and the rest of the assessed methods.

Dataset	Deep Feature-based Methods					Perceptual Hashing Methods			
	Dlib	VGG16	OpenFace	FaceNet	ArcFace	SS-S-SF	RP-IVD	pHash	OSF-DNS
	[63]	[62]	[7]	[11]	[5]	[46]	[60]	[61]	
LFW	35.82	31.56	23.20	36.09	5.17	43.21	44.25	84.30	**91.59**
CFPW	40.28	46.08	18.67	51.58	3.50	58.64	23.33	73.25	**75.54**
MEDS-II	67.12	53.16	24.75	31.01	6.24	60.21	21.92	99.23	**99.46**
NIMH-ChEFS	98.75	42.10	82.52	55.60	7.49	56.22	33.81	**99.37**	**99.37**
CUHK	94.14	60.56	94.68	66.48	5.68	62.75	36.70	98.93	**99.46**
VGGFace2	39.42	24.27	13.84	19.30	3.38	32.95	25.03	65.18	**67.24**

verification experiment, i.e., the one explained in Section 6.1.1. The results are reported in Table 3.

In this experiment, OSF-DNS achieves the highest face verification performance for both resolutions in LFW and CFPW datasets. Nonetheless, the behavior with respect to the resolutions is different depending on the dataset. In the LFW dataset, the accuracy is higher with the 90 × 90 resolution whereas in CFPW the performance is higher when the images were resized to 180 × 180. Moreover, this happens for all the assesed perceptual hashing methods. To gain some insight of the reason for this behavior, we computed the average resolution of the cropped faces, which is around 110 × 90 in LFW, and 209 × 184 in CFPW. Therefore, the lower the mismatch between the original and the resized sizes, the better the verification performance appears to be.

Note that we have excluded the deep feature-based methods, i.e., Dlib Face Recognition [63], VGG16 [62], OpenFace [7], FaceNet [11], and ArcFace [5] from Table 3 because they obtained the same results that those reported in Table 2. Indeed, each of the deep feature-based face recognition methods requires the input image to have a

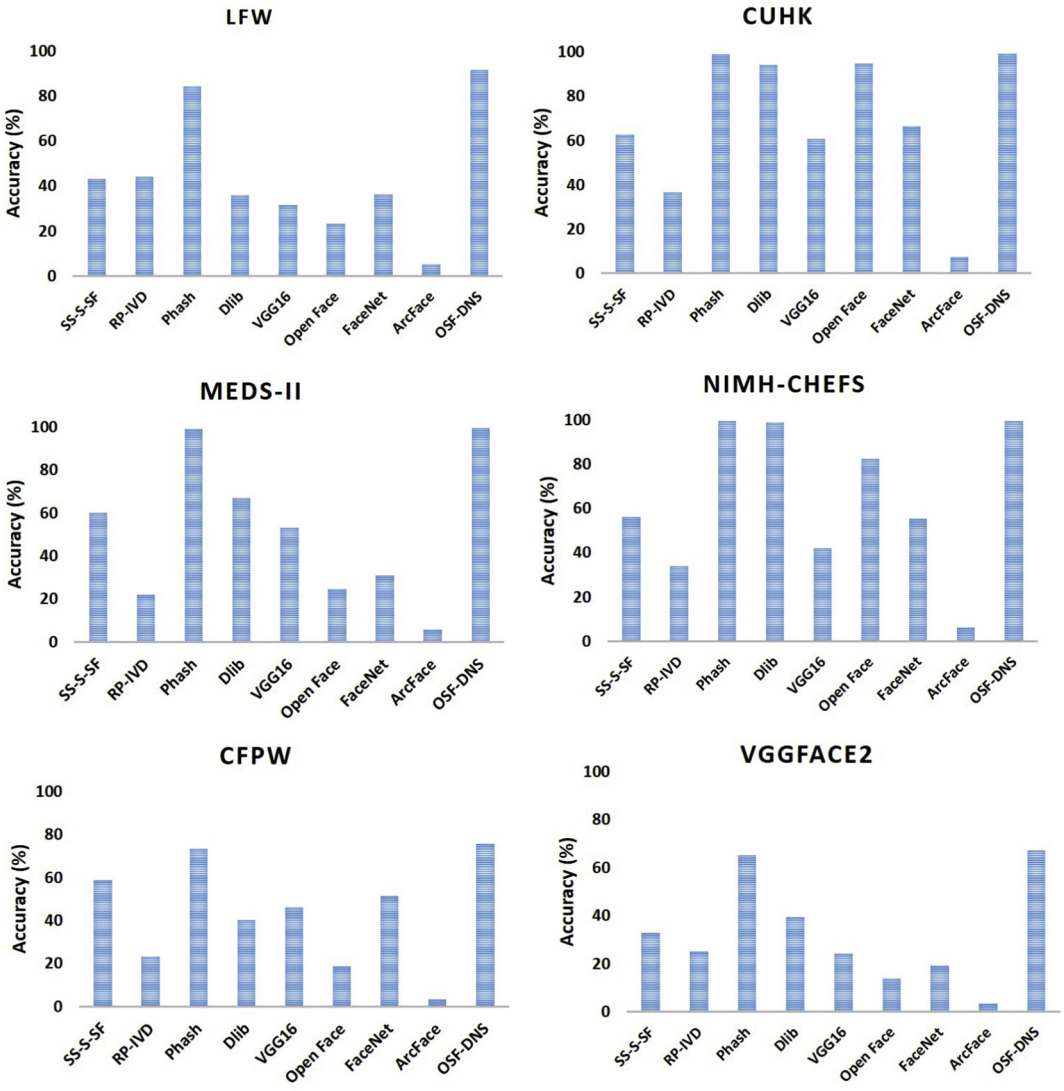

Fig. 7. Face verification accuracy (in %) against the AERO attack OSF-DNS and the rest of the assessed methods for the datasets LFW, CUHK, MEDS-II, NIMH-CHEFS, CFPW and VGGFace2.

Table 3
The effect of face image resolution on the performance of face verification (accuracy in %) against the AERO attack using OSF-DNS and other perceptual hashing methods.

Dataset	Resolution	SS-S-SF [46]	RP-IVD [60]	pHash [61]	OSF-DNS
LFW	90 × 90	47.90	46.86	89.72	**92.15**
	180 × 180	42.80	27.43	71.25	**85.98**
CFPW	90 × 90	23.20	7.32	63.10	**68.90**
	180 × 180	68.50	44.00	82.06	**86.60**

fixed size, so, even though we resized the images to any size different from 120 × 120, we would have to resize them again to fit them into each architecture.

In order to further explore face verification performance against the AERO attack, we carried out another experiment. Specifically, we wanted to assess how robust was OSF-DNS against different sizes of the occlusion mask. To assess that, we applied four different occlusion sizes to each image, i.e. 10%, 20%, 30%, and 40% of the image height, to build four different AERO attacked face datasets of LFW. Fig. 8 shows an example of the different occlusion ratios. Then, we carried out the same experiment mentioned above, for each of the four occlusion ratios. In this experiment, we compared OSF-DNS with pHash and FaceNet because they attained the highest performance among the perceptual hashing and the CNN-based approaches, respectively (see Table 2). Fig. 9 depicts the accuracies for the four different occlusion ratios. It is clear that the perceptual hashing-based methods outperformed FaceNet for different occlusion ratios, being OSF-DNS better in all cases. The verification accuracy is worse for the ratio of 40% because it covers most of the information of a face, especially the eyes and nose. Fig. 9 also shows that the accuracy decreased with respect to the occlusion ratio.

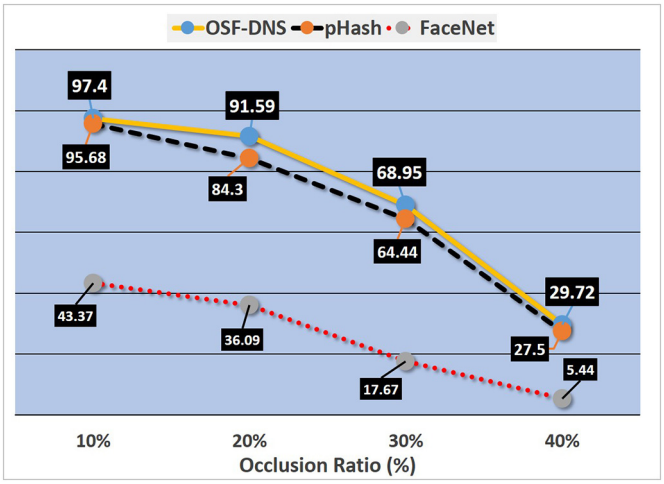

Fig. 9. Accuracy (%) with respect to different occlusion ratios, i.e., 10%, 20%, 30% and 40% using OSF-DNS, pHash and FaceNet for face verification robust to occlusions on the LFW dataset.

6.2.2. Face classification

In the face classification experiments, we followed the procedure described in Section 5.2 to evaluate the LFW and CFPW datasets. Table 4 and Fig. 10 show the results of this classification experiment.

In the case of AERO attack in the LFW Face Database, we observe that OSF-DNS obtains an accuracy of 82.74%, 45.34%, 89.53%, 88.14% and 86.06% using the Linear, RBF, Poly-2, Poly-4, and Poly-6 kernel functions, respectively. This outperforms the results obtained by the features of the three other hashing approaches as well as the deep learning-based assessed methods.

In the case of the CFPW face dataset, SS-S-SF obtains the highest classification accuracy, i.e., 51.06%, with Linear kernel function, and FaceNet attains the highest classification accuracy of 54,14% for RBF. On the contrary, pHash attains 52,32% classification accuracy for Poly-2 kernel function which is higher compared to the rest of the approaches. OSF-DNS, obtains the best performance overall since, as can be seen in Fig. 10, it attains accuracies of 58.6% and 63.24% for the kernel functions Poly-4 and Poly-6, respectively.

Similarly to what we did for the face verification (see Section 6.2.1), we have carried out an additional experiment to assess the impact of the resizing of the face image on the performance of the classification. Specifically, we have repeated the classification experiment explained in Section 6.1.2 but resizing the cropped images to 90 × 90 and 180 × 180. Table 5 shows the AERO attack face classification results for both resolutions.

It can be observed that for LFW, OSF-DNS obtains the highest classification accuracies for all the kernel functions with the 90 × 90 resolution. OSF-DNS obtains the highest accuracies for all the kernels with the 180 × 180 resolution, except for RBF and Poly-2, in which pHash attains 69.91% and 67.32%, respectively. In contrast, for CFPW, OSF-DNS attains the highest classification accuracies for all the kernel functions, except for Poly-2 and Poly-6, in which pHash performed better, with the input face resolution of 90 × 90. Subsequently, the performance of OSF-DNS is higher for Linear, RBF, and Poly-6 kernels, with the input face resolution of 180 × 180.

Similarly to what we observed in face verification (see Section 6.2.1), we noticed that the classification performance is better when the resolution of the resized cropped face image is close to the original resolution. Note that we have excluded the deep-feature based methods, i.e., Dlib Face Recognition [63], VGG16 [62], OpenFace [7], FaceNet [11], and ArcFace [5] from this experiment for the same reason that we reported in face verification task (see in Section 6.2.1).

6.3. Performance analysis for other facial occlusions

We carried out another experiment to evaluate the face verification and classification performance against other facial occlusions, such as the nose or mouth occlusion. More precisely, we wanted to evaluate the robustness of our proposal against the occlusion over the mouth and mouth + nose regions of the face, respectively. To do that, we

Fig. 8. Example of the AERO attack face image with five different occlusion ratios, i.e., 10%, 20%, 30%, and 40%, from NIMH-ChEFS dataset.

Table 4
Accuracy (in %) of the face classification against the AERO attack obtained by SVM with different kernel functions using OSF-DNS and the rest of the assessed methods as features.

Dataset	Kernels	Deep Feature-based Methods					Perceptual Hashing Methods			
		Dlib	VGG16	OpenFace	FaceNet	ArcFace	SS-S-SF	RP-IVD	pHash	OSF-DNS
		[63]	[62]	[7]	[11]	[5]	[46]	[60]	[61]	
LFW	Linear	4.00	12.67	5.3	13.45	6.58	28.01	23.81	77.08	**82.74**
	RBF	9.36	4.81	11.2	4.53	4.01	8.22	17.31	74.66	**45.34**
	Poly-2	19.25	4.81	21.5	14.23	6.82	16.28	12.42	74.34	**89.53**
	Poly-4	27.39	4.89	9.5	20.02	4.33	14.98	7.02	72.80	**88.14**
	Poly-6	23.56	4.54	5.3	20.02	2.78	12.86	7.08	72.16	**86.06**
CFPW	Linear	48.78	14.56	37.42	50.94	3.23	**51.06**	19.30	28.98	26.00
	RBF	50.94	7.82	2.00	**54.14**	2.00	43.74	13.78	41.32	18.18
	Poly-2	46.82	6.82	29.38	49.26	3.75	32.52	19.16	**52.32**	38.52
	Poly-4	31.02	3.01	10.56	22.06	2.33	28.86	19.32	55.10	**58.60**
	Poly-6	17.28	1.29	3.92	6.68	2.05	24.80	19.32	61.54	**63.24**

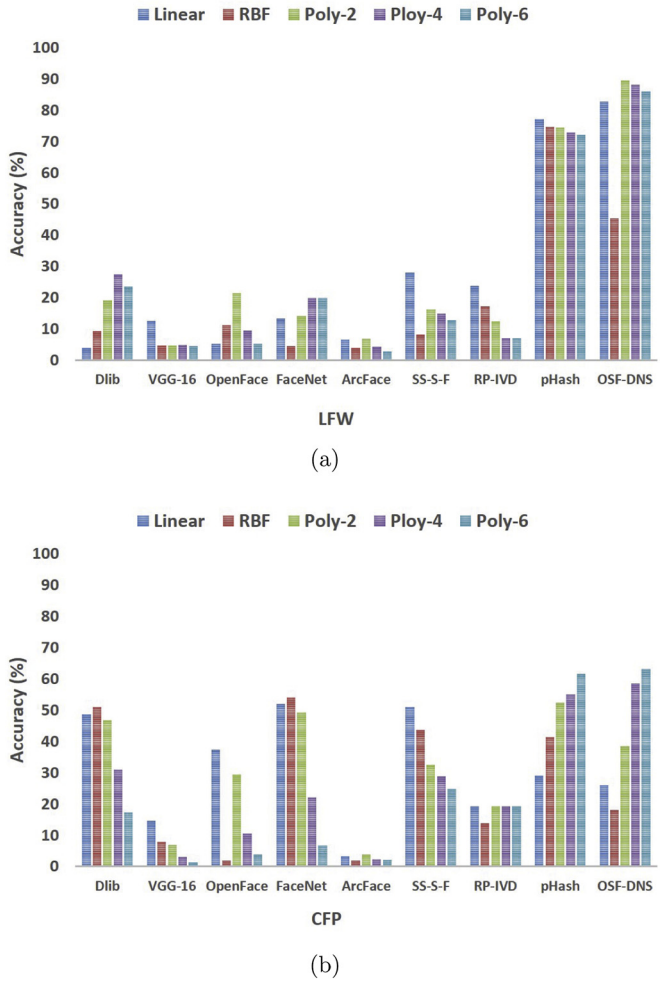

Fig. 10. a) and b) show the performance of OSF-DNS, other perceptual hashing features obtained with SS-S-SF [46], RP-IVD [60] and pHash [61], and deep learning-based features obtained by Dlib Face Recognition [63], VGG16 [62], OpenFace [7], FaceNet [11], and ArcFace [5], for occluded face classification on LFW and CFPW datasets, respectively.

built (a) mouth and (b) mouth and nose occluded versions of the face dataset of LFW using an approach similar to the one mentioned in Section 3. In this case, we have used the points of the mouth detected by the face detector instead of the external points in the eyes as P_1 and P_2 (see Section 3) and then, we followed the same steps explained in Section 3, changing the width of the mask. Fig. 11 shows some examples of mouth and mouth + nose occluded faces. Then, we carried out

Table 5
The effect of face image resolution on the performance of face classification (accuracy in %) against the AERO attack using OSF-DNS and other perceptual hashing methods.

Dataset	Resolution	Kernels	SS-S-SF	RP-IVD	pHash	OSF-DNS
			[46]	[60]	[61]	
LFW	90 × 90	Linear	39.40	19.24	81.78	**84.24**
		RBF	7.70	16.38	71.07	**45.10**
		Poly-2	15.70	13.18	71.46	**66.58**
		Poly-4	12.00	7.18	71.20	**76.65**
		Poly-6	27.40	7.24	69.78	**84.34**
	180 × 180	Linear	41.10	19.42	64.73	**77.68**
		RBF	17.80	20.25	**69.91**	43.67
		Poly-2	17.70	20.23	**67.32**	66.00
		Poly-4	12.5	7.44	71.11	**72.81**
		Poly-6	41.10	7.42	70.73	**81.50**
CFPW	90 × 90	Linear	9.00	3.42	30.08	**33.11**
		RBF	9.00	2.38	17.36	**21.42**
		Poly-2	13.20	3.46	**38.82**	36.45
		Poly-4	13.00	6.48	45.18	**55.74**
		Poly-6	19.87	5.42	**65.74**	61.08
	180 × 180	Linear	45.50	30.00	38.92	**78.38**
		RBF	24.00	32.00	20.08	**34.24**
		Poly-2	**38.10**	35.00	38.60	33.52
		Poly-4	**46.40**	43.30	15.16	45.68
		Poly-6	57.50	43.90	63.72	**73.58**

the same experimental procedures, mentioned in Section 6.1), to evaluate the face verification and classification performances using both occluded versions.

It can be observed in Table 6 that the verification accuracy of OSF-DNS is comparatively higher (i.e. 97.60%) than the rest of the assessed methods when the occlusion appears only in the mouth area. We also noticed that the verification accuracies of deep feature-based methods also improves significantly. It might be because the most useful information that these approaches capture comes from the eye region. In the case of face classification, OSF-DNS obtains the best results with accuracies of 95.00%, 89.60%, 97.10%, 96.60%, and 93.30% for Linear, RBF, Poly-2, Poly-4, and Poly-6 kernel functions, respectively.

On the contrary, when occlusion appears on the mouth and nose together, Table 7 shows that the deep learning-based features obtained the highest verification accuracies (being the best one VGG16 with 89.60%). With respect to the face classification, besides, FaceNet obtains the highest classification accuracies for all the kernel functions, except for Poly-4, and Poly-6 in which Dlib and OSF-DNS are the best ones.

To sum up, the face verification and classification performance of all the assessed methods, OSF-DNS included, improves significantly when the occlusion appears in the mouth region compared to when occlusion appears over the eye region. Besides, the perceptual hashing works

Fig. 11. Examples of occlusion a) over mouth (bottom first column) of some faces (top first column), and b) over mouth and nose (bottom second column) of some faces (top second column) from LFW dataset.

Table 6
Face verification (FV) and classification (FC) accuracies (in %) against face occlusion over the mouth obtained by OSF-DNS and the rest of the assessed methods on LFW dataset.

Exp. Type	Kernels	Deep Feature-based Methods					Perceptual Hashing Methods			
		Dlib	VGG16	OpenFace	FaceNet	ArcFace	SS-S-SF	RP-IVD	pHash	OSF-DNS
		[63]	[62]	[7]	[11]	[5]	[46]	[60]	[61]	
FV	–	85.80	95.00	67.75	93.40	60.70	81.50	45.50	96.25	**97.60**
FC	Linear	72.90	80.70	57.70	87.30	59.00	79.40	37.40	94.40	**95.00**
	RBF	23.80	23.90	40.70	38.20	33.80	54.10	27.70	53.90	**89.60**
	Poly-2	52.25	70.90	60.45	47.70	75.25	90.10	47.90	89.50	**97.10**
	Poly-4	42.40	50.60	62.00	45.40	38.00	90.40	49.60	**96.60**	**96.60**
	Poly-6	32.50	37.00	66.72	37.00	23.70	90.60	52.00	92.90	**93.30**

worse than the deep learning-based methods in face verification when occlusion appears in the mouth and nose regions together. This is also happening in face classification, except for the Poly-6 kernel function. Deep-learning-based methods work better when the eyes are not occluded since it appears that most of the information they extract from the face lies in the eye region. In contrast, perceptual hashing methods perform better when the occlusion area is not too big. However, the bigger the occluded area, the lower their performance is.

6.4. Computational time analysis

We analyzed and compared the time complexity of the proposed hashing method OSF-DNS with the assessed deep feature-based methods and perceptual hashing approaches. To perform that, we randomly selected 1000 images from the LFW dataset with a resolution of 250 × 250 pixels. Then, we measured the average time to (a) detect and crop the face from an input image and then (b) the average time required to extract the hash code/embedding of the face image. Later, we also reported (c) the retrieval time required to verify an AERO attack face from the dataset. We did the same for the face classification task where we only considered the Poly-6 kernel function, since it is the most time-consuming among the ones assessed in this paper. All methods have been run on a PC with 128GB RAM and CPU Intel (R) Core(TM) i7.

We found that the average time to detect and crop a face image using the face detection method MTCNN was 1.17 seconds. The average time of the other steps (i.e. computation of the hash or embedding, verification and classification), which depended on each specific method, are reported in Table 8. With respect to the computation of the hash codes or the embeddings, the fastest method for computing the hash code is pHash, which required an average time of 0.005 seconds per face image, whereas OSF-DNS required 0.009 seconds. On the contrary, SS-S-SF required more computational time than the rest of the perceptual hashing and deep feature-based methods. Among the deep learning-based methods, VGG16 was the slowest in the extraction of the embeddings of a face, i.e., 0.856 seconds.

With respect to the average computational time to verify an AERO attacked face, i.e., measuring the similarity between the hash code of an attacked face and the hash codes of all the original. In this scenario, RP-IVD required 2.812 seconds, which is slightly lower than OSF-DNS

Table 7
Face verification (FV) and classification (FC) accuracies (in %) against face occlusion over the mouth and nose obtained by OSF-DNS and the rest of the assessed methods on LFW dataset.

Exp. Type	Kernels	Deep Feature-based Methods					Perceptual Hashing Methods			
		Dlib	VGG16	OpenFace	FaceNet	ArcFace	SS-S-SF	RP-IVD	pHash	OSF-DNS
		[63]	[62]	[7]	[11]	[5]	[46]	[60]	[61]	
FV	–	32.25	**89.60**	37.09	79.84	25.80	17.20	13.27	32.42	32.63
FC	Linear	33.90	45.05	25.00	**54.00**	12.14	19.09	10.15	11.09	13.10
	RBF	17.10	25.21	13.80	**26.07**	7.10	9.74	5.25	16.33	15.83
	Poly-2	30.50	26.46	7.35	**29.13**	10.40	9.64	10.10	20.34	20.50
	Poly-4	**27.90**	21.87	7.35	16.23	8.50	15.18	12.02	27.35	25.78
	Poly-6	19.70	16.04	11.62	9.26	5.10	24.89	17.90	29.85	**30.45**

Table 8
Average computational time (in seconds) to compute the hash or the face embedding (CH), and to verify (FV) or classify (FC) a face image obtained by OSF-DNS and the five assessed deep feature-based approachesapplied on images from LFW dataset.

Analysis situation	Deep Feature-based Methods					Perceptual Hashing Methods			
	Dlib [63]	VGG16 [62]	OpenFace [7]	FaceNet [11]	ArcFace [5]	SS-S-SF [46]	RP-IVD [60]	pHash [61]	OSF-DNS
CH	0.397	0.856	0.768	0.743	0.794	45.117	0.825	0.005	0.009
FV	7.110	132.110	6.000	20.770	44.610	4.190	2.812	3.430	3.191
FC (Poly-6)	3.640	35.681	2.400	4.900	7.950	0.871	0.503	0.491	0.370

(i.e., 3.191 s). In contrast, VGG16 took 132.110 seconds, much higher compared to the rest of the methods. In the case of the classification of the attacked face, OSF-DNS required 0.370 seconds, whereas VGG16 required 35.681 seconds.

All in all, OSF-DNS achieved the best total computational time (i.e., including the computation of the hash code and the time to verify or classify an AERO attacked face): 4.37 and 1.549 seconds, respectively, which are lower than pHash, which required 4.605 and 1.666 seconds, respectively (see Fig. 12). In contrast, the rest of the assessed methods, i.e., the perceptual hashing methods, SS-Salient-SF, RP-IVD as well as the deep feature-based methods, i.e., Dlib, VGG16, OpenFace, FaceNet, and ArcFace, required higher computational times.

IIn conclusion, due to the excellent results on the AERO attacked face verification and classification tasks plus its good performance in terms of computational time, OSF-DNS can be recommended for a system to accurately classify or verify an AERO attacked face from CSEM.

7. Conclusion and future work

In this work, we proposed a new perceptual hashing scheme, named One-Shot Frequency Dominant Neighborhood Structure (OSF-DNS). We have applied it for face verification and classification against the AERO attack, which may be very helpful to LEAs in cases involving Child Sexual Exploitation Material (CSEM). LEAs have evidence databases containing offenders or victims' faces obtained from other cases. However, it is common to find CSEM in which faces are occluded by the offenders distributing the material, most times in the eye region. Therefore, retrieving the non-attack faces by matching the attack faces or classifying the occluded faces to get their identities may be helpful for finding evidences in cases involving CSEM.

The extraction of OSF-DNS takes advantage of the global texture energy and local Dominant Neighborhood Structure of an image. As any other perceptual hashing method, it does not require any prior training for face verification. In the classification experiment, we have used OSF-DNS to describe the AERO attacked faces with the goal of obtaining their identity. First, an SVM classifier has been trained with the OSF-DNS features extracted from non-occluded faces. Then, the attacked faces, also described by OSF-DNS, were classified by this SVM. The objective was to simulate the real situations in which the LEAs only have non-occluded faces in their databases and want to get the identity of a person whose face has been occluded in a new image. Due to the lack of publicly available datasets containing AERO attacked faces, we also presented an automatic procedure to create an AERO attacked face dataset.

We compared the performance of OSF-DNS for face verification with three other perceptual hashing methods: RP-IVD, SS-S-SF, and pHash, and also with the features obtained from five deep learning-based techniques: Dlib, VGG16, OpenFace, FaceNet, and ArcFace. We employed six different datasets, LFW, CFPW, MEDS-II, NIMH-ChEFS, CUHK, and VGGFace2, together with their six AERO versions. The proposed hashing method obtained the highest accuracy in the task of occluded face verification for all datasets. In the case of the AERO attack face classification we compared OSF-DNS with the features obtained from the perceptual hashing methods (i.e. SS-S-SF, RP-IVD, and pHash), five deep learning techniques: Dlib, VGG16, OpenFace, FaceNet, and ArcFace over two labeled datasets, i.e., LFW and CFPW. The results demonstrate that our proposal achieves the highest accuracy with almost all the kernel functions. These results show that our proposed scheme can be recommended for forensic tools to verify whether an AERO attack

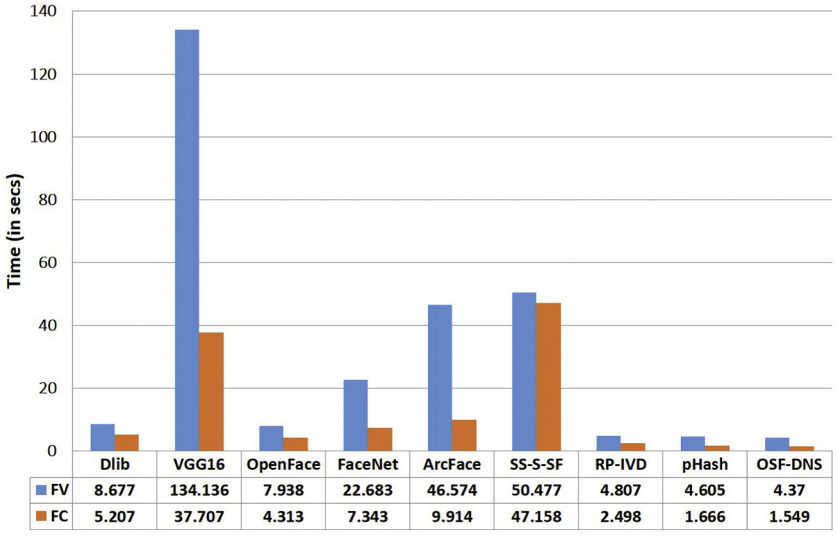

Fig. 12. Average (in total) computational time (in seconds) to verify and classify an AERO attack face using OSF-DNS and other perceptual hashing methods plus deep-feature based approaches.

face is found in a database of non-attack faces. This may be helpful to make a legal case involving CSEM or other criminal materials.

In future works, we will tackle the problems of occluded face recognition without prior training with the non-occluded faces. We will also study the problem of face verification, classification, or recognition under adversarial attacks of other parts of the face.

Declaration of Competing Interest

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

Acknowledgments

This research is supported by the framework agreement between the University of León and (INCIBE Spanish National Cybersecurity Institute) under Addendum 01. We also gratefully acknowledge the support of Nvidia Corporation for their kind donation of GPUs (GeForce GTX Titan X and K-40) that were used in this work. Finally, we would like to thank Professor Zhenjun Tang for sharing with us the implementation of his perceptual hashing method Ring Partition and Invariant Vector Distance and Professor Chuan Qin for sharing the implementation of his method Selective Sampling for Salient Structure Features.

References

[1] D. Zeng, F. Zhao, S. Ge, W. Shen, Fast cascade face detection with pyramid network, Pattern Recogn. Lett. 119 (2019) 180–186.
[2] T. Lu, Q. Zhou, W. Fang, Y. Zhang, Discriminative metric learning for face verification using enhanced Siamese neural network, Multim. Tools Applic. 80 (2021) 8563–8580.
[3] E.-J. Cheng, K.-P. Chou, S. Rajora, B.-H. Jin, M. Tanveer, C.-T. Lin, K.-Y. Young, W.-C. Lin, M. Prasad, Deep sparse representation classifier for facial recognition and detection system, Pattern Recogn. Lett. 125 (2019) 71–77.
[4] A. Jourabloo, X. Liu, Large-pose face alignment via CNN-based dense 3D model fitting, IEEE conference on computer vision and pattern recognition 2016, pp. 4188–4196.
[5] D. Jiankang, G. Jia, X. Niannan, Z. Stefanos, ArcFace: additive angular margin loss for deep face recognition, Computer Vision and Pattern Recognition (CVPR) 2019, pp. 4690–4699.
[6] S.M. Iranmanesh, B. Riggan, S. Hu, N.M. Nasrabadi, Coupled generative adversarial network for heterogeneous face recognition, Image Vis. Comput. 94 (2020) 1–10.
[7] B. Amos, B. Ludwiczuk, M. Satyanarayanan, Openface: A General-purpose Face Recognition Library with Mobile Applications, CMU School of Computer Science, 2016.
[8] F.V. Massoli, G. Amato, F. Falchi, Cross-resolution learning for face recognition, Image Vis. Comput. 99 (2020) 1–15.
[9] R. Zhou, S.K. Roy, M. Harandi Fang, L. Petersson, Cross-correlated attention networks for person re-identification, Image Vis. Comput. 100 (2020) 1–8.
[10] L. Best-Rowden, H. Han, C. Otto, B.F. Klare, A.K. Jain, Unconstrained face recognition: identifying a person of interest from a media collection, IEEE Trans. Inform. Foren. Security 9 (12) (2014) 2144–2157.
[11] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, Proceedings of the IEEE conference on computer vision and pattern recognition 2015, pp. 815–823.
[12] F. Vakhshiteh, R. Ramachandra, A. Nickabadi, Threat of Adversarial Attacks on Face Recognition: A Comprehensive Survey, 2020 ArXiv abs/2007.11709.
[13] J. Mathai, I. Masi, W. AbdAlmageed, Does generative face completion help face recognition? International Conference on Biometrics (ICB) 2019, pp. 1–8.
[14] P. Görgel, A. Simsek, Face recognition via deep stacked Denoising sparse autoencoders (DSDSA), Appl. Math. Comput. 355 (2019) 325–342.
[15] J. Dong, L. Zhang, Y. Chen, W. Jiang, Occlusion expression recognition based on non-convex low-rank double dictionaries and occlusion error model, Signal Process. Image Commun. 76 (2019) 81–88.
[16] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[17] P. Zhang, X. You, W. Ou, C.P. Chen, Y.-m. Cheung, Sparse discriminative multi-manifold embedding for one-sample face identification, Pattern Recogn. 52 (2016) 249–259.
[18] B.F. Klare, A.K. Jain, Heterogeneous face recognition using kernel prototype similarities, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1410–1422.
[19] A. Gangwar, E. Fidalgo, E. Alegre, V. González-Castro, Pornography and child sexual abuse detection in image and video: A comparative evaluation, 8th International Conference on Imaging for Crime Detection and Prevention (ICDP 2017) 2017, pp. 37–428.
[20] R. Biswas, V. González-Castro, E. Fidalgo Fernández, D. Chaves, Boosting child abuse victim identification in Forensic Tools with hashing techniques, V Jornadas Nacionales de Investigación en Ciberseguridad (JNIC) 2019, pp. 344–345.
[21] M. Schneider, S.-F. Chang, A robust content based digital signature for image authentication, Proceedings of 3rd IEEE International Conference on Image Processing, Vol. 3, 1996, pp. 227–230.
[22] W. Lu, M. Wu, Multimedia forensic hash based on visual words, in: Image processing (ICIP), 17th IEEE international conference on, IEEE 2010, pp. 989–992.
[23] C.-S. Lu, C.-Y. Hsu, Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication, Multimedia Systems 11 (2) (2005) 159–173.
[24] L. Xie, L. Zhu, P. Pan, Y. Lu, Cross-modal self-taught hashing for large-scale image retrieval, Signal Process. 124 (2016) 81–92.
[25] R. Biswas, V. González-Castro, E. Fidalgo, E. Alegre, Perceptual image hashing based on frequency dominant neighborhood structure applied to Tor domains recognition, Neurocomputing 27 (2020) 778–790.
[26] Y. Zhong, W. Deng, J. Hu, D. Zhao, X. Li, D. Wen, SFace: sigmoid-constrained hypersphere loss for robust face recognition, IEEE Trans. Image Process. 30 (2021) 2587–2598.
[27] S. Bharadwaj, H.S. Bhatt, M. Vatsa, R. Singh, Domain specific learning for newborn face recognition, IEEE Trans. Inform. Forens. Security 11 (7) (2016) 1630–1641.
[28] L. Zheng, K. Idrissi, C. Garcia, S. Duffner, A. Baskurt, Triangular similarity metric learning for face verification, 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Vol. 1, 2015, pp. 1–7.
[29] A.M. Andrés, S. Padovani, M. Tepper, J. Jacobo-Berlles, Face recognition on partially occluded images using compressed sensing, Pattern Recogn. Lett. 36 (2014) 235–242.
[30] M. Koç, A novel partition selection method for modular face recognition approaches on occlusion problem, Mach. Vis. Appl. 32 (35) (2021) 1–11.
[31] N. Dagnes, F. Marcolin, F. Nonis, S. Tornincasa, E. Vezzetti, 3D geometry-based face recognition in presence of eye and mouth occlusions, Int. J. Interact. Des. Manuf. 13 (2019) 1617–1635.
[32] M. Domingo, B. Kevin, Face recognition via adaptive sparse representations of random patches, IEEE International Workshop on Information Forensics and Security 2014, pp. 13–18.
[33] W. Cho-Ying, D. Jian-Jiun, Occlusion pattern-based dictionary for robust face recognition, Proceedings of the IEEE International Conference Multimedia and Expo 2016, pp. 1–6.
[34] M.A. Mustafa, P. Nadith, L. Wanquan, L. Ling, Face recognition against occlusions via colour fusion using 2D-MCF model and SRC, Pattern Recogn. Lett. 95 (2017) 14–21.
[35] L. Shengcai, K.J. Anil, Z.L. Stan, Partial face recognition: an alignment-free approach, IEEE Trans. Pattern Anal. Mach. Intell. 35 (5) (2013) 1193–1205.
[36] D. Yueqi, L. Jiwen, F. Jianjiang, Z. Jie, Topology preserving structural matching for automatic partial face recognition, IEEE Trans. Inform. Forens. Security 13 (7) (2018) 1823–1837.
[37] W. Ou, X. Luan, J. Gou, Q. Zhou, W. Xiao, X. Xiong, W. Zeng, Robust discriminative nonnegative dictionary learning for occluded face recognition, Pattern Recogn. Lett. 107 (2018) 41–49.
[38] F. Zhao, J. Feng, J. Zhao, W. Yang, S. Yan, Robust LSTM-autoencoders for face de-occlusion in the wild, IEEE Trans. Image Process. 27 (2018) 778–790.
[39] Y. Xu, L. Zhao, F. Qin, Dual attention-based method for occluded person re-identification, Knowl.-Based Syst. 212 (2021) 106554.
[40] G. Guangwei, Y. Jian, J. Xiao-Yuan, S. Fumin, Y. Wankou, Y. Dong, Learning robust and discriminative low-rank representations for face recognition with occlusion, Pattern Recogn. 66 (2017) 129–143.
[41] Y. Cho, J. Wu, D. Jiun, Occluded face recognition using low-rank regression with generalized gradient direction, Pattern Recogn. 80 (2018) 256–268.
[42] Y. Yu-Feng, D. Dao-Qing, R. Chuan-Xian, H. Ke-Kun, Discriminative multi-scale sparse coding for single-sample face recognition with occlusion, Pattern Recogn. 66 (2017) 302–312.
[43] L. Yang, Z. Fan, S. Ling, H. Junwei, Face recognition with a small occluded training set using spatial and statistical pooling, Inf. Sci. 430–431 (2018) 634–644.
[44] Z. Tang, Y. Yu, H. Zhang, M. Yu, C. Yu, X. Zhang, Robust image hashing via visual attention model and ring partition, Math. Biosci. Eng. 16 (2019) 6103–6120.
[45] C. Qin, Y. Hu, H. Yao, X. Duan, L. Gao, Perceptual image hashing based on weber local binary pattern and color angle representation, IEEE Access 7 (2019) 45460–45471.
[46] C. Qin, X. Chen, J. Dong, X. Zhang, Perceptual image hashing with selective sampling for salient structure features, Displays 45 (2016) 26–37.
[47] R. Biswas, E. Fidalgo, E. Alegre, Recognition of service domains on TOR dark net using perceptual hashing and image classification techniques, 8th International Conference on Imaging for Crime Detection and Prevention (ICDP) 2017, pp. 13–15.
[48] E. Fidalgo Fernández, E. Alegre Gutiérrez, L. Fernández Robles, V. González Castro, Early fusion of multi-level saliency descriptors for image classification, RIAI Rev. Iberoam. Autom. Inform. Ind. 16 (3) (2019).
[49] R. Davarzani, S. Mozaffari, K. Yaghmaie, Perceptual image hashing using center-symmetric local binary patterns, Multimed. Tools Appl. 75 (8) (2016) 4639–4667.
[50] Z. Tang, X. Li, X. Zhang, S. Zhang, Y. Dai, Image hashing with color vector angle, Neurocomputing 308 (2018) 147–158.
[51] X. Yuan, Y. Zhao, Perceptual image hashing based on three-dimensional global features and image energy, IEEE Access 9 (2021) 49325–49337.
[52] W. Fang, H.-M. Hu, Z. Hu, S. Liao, B. Li, Perceptual hash-based feature description for person re-identification, Neurocomputing 272 (2018) 520–531.
[53] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Process. Lett. 23 (10) (2016) 1499–1503.
[54] B.H. Gary, M. Marwan, B. Tamara, L.-M. Erik, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Tech. Rep. 7–49 Univ. Massachusetts, USA, 2007.

[55] P.F. Andrew, O. Nick, G. Whiddon, I.W. Craig, NIST Special Databse 32 - Multiple Encounter Dataset II (MEDS-II), Tech. Rep. 7807 National Institute of Standards and Technology (NIST), 2011.
[56] S. Sengupta, J. Chen, C. Castillo, V.M. Patel, R. Chellappa, D.W. Jacobs, Frontal to profile face verification in the wild, IEEE Winter Conference on Applications of Computer Vision (WACV) 2016, pp. 1–9.
[57] L.E. Helen, S.P. Daniel, N. Eric, L. Ellen, E. Monique, E.T. Kenneth, A. Adrian, The NIMH child emotional faces picture set (NIMH-ChEFS): a new set of children's facial emotion stimuli, Int. J. Methods Psychiatr. Res. 20 (3) (2011) 145–156.
[58] F.M. Khellah, Texture classification using dominant neighborhood structure, IEEE Trans. Image Process. 20 (11) (2011) 3270–3279.
[59] A. Bulat, G. Tzimiropoulos, How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks), 2017 IEEE International Conference on Computer Vision (ICCV) 2017, pp. 1021–1030.
[60] Z. Tang, X. Zhang, X. Li, S. Zhang, Robust image hashing with ring partition and invariant vector distance, IEEE Trans. Inform. Forens. Security 11 (1) (2016) 200–214.
[61] C. Zauner, Implementation and Benchmarking of Perceptual Image Hash Functions, Master's thesis University of Applied Sciences Hagenberg, Austria, 2010.
[62] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations 2015, pp. 1–14.
[63] S. Sharma, S. Karthikeyan, K.R. Sathees, FAREC – CNN based efficient face recognition technique using Dlib, International Conference on Advanced Communication Control and Computing Technologies (ICACCCT) 2016, pp. 192–195.