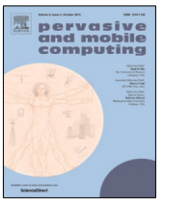
Pervasive and Mobile Computing 92 (2023) 101801

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

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



BLUFADER: Blurred face detection & recognition for privacy-friendly continuous authentication☆

Matteo Cardaioli [a,c], Mauro Conti [a,e], Gabriele Orazi [a,d], Pier Paolo Tricomi [a,*], Gene Tsudik [b]

[a] Department of Mathematics, University of Padua, Padua, Italy
[b] Department of Computer Science, University of California, Irvine, USA
[c] GFT Italy, Milan, Italy
[d] FDM Business Services, Milan, Italy
[e] Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Article history:
Received 12 September 2022
Received in revised form 31 March 2023
Accepted 28 April 2023
Available online 10 May 2023

Keywords:
Continuous authentication
De-authentication
Lunchtime attacks
Privacy
Usability
Deep learning
Blurred face detection

ABSTRACT

Authentication and de-authentication phases should occur at the beginning and end of secure user sessions, respectively. A secure session requires the user to pass the former, but the latter is often underestimated or ignored. Unattended or dangling sessions expose users to well-known *Lunchtime Attacks*. To mitigate this threat, researchers focused on automated de-authentication systems, either as a stand-alone mechanism or as a result of continuous authentication failures. Unfortunately, no single approach offers security, privacy, and usability. Face-recognition methods, for example, may be suitable for security and usability, but they violate user privacy by continuously recording their actions and surroundings.

In this work, we propose BLUFADER, a novel continuous authentication system that takes advantage of blurred face detection and recognition to fast, secure, and transparent de-authenticate users, preserving their privacy. We obfuscate a webcam with a physical blur layer and use deep learning algorithms to perform face detection and recognition continuously. To evaluate BLUFADER's practicality, we collected two datasets formed by 30 recruited subjects (users) and thousands of physically blurred celebrity photos. The de-authentication system was trained and evaluated using the former, while the latter was used to appraise the privacy and increase variance at training time. To guarantee the privacy-preserving effectiveness of the selected physical blurring filter, we show that state-of-the-art deblurring models are not able to revert our physical blur. Further, we demonstrate that our approach outperforms state-of-the-art methods in detecting blurred faces, achieving up to 95% accuracy. Moreover, BLUFADER effectively de-authenticates users up to 100% accuracy in under 3 seconds, while satisfying security, privacy, and usability requirements. Last, our continuous authentication face recognition module based on Siamese Neural Network preventively protect users from adversarial attacks, enhancing the overall system security.

☆ The manuscript is the extended version of Cardaioli et al. (2022), published at PerCom 2022.

* Corresponding author.
E-mail address: tricomi@math.unipd.it (P.P. Tricomi).

https://doi.org/10.1016/j.pmcj.2023.101801

1. Introduction

Users must authenticate themselves before accessing any modern computing device (such as a desktop computer, workstation, laptop, tablet, or smartphone). A user typically must demonstrate one or more of the following during the authentication process: (1) a password or PIN, (2) a biometric, such as a face or fingerprint, and (3) a device, such as a secure dongle or smartphone. Secure means of authentication have been created and supported through massive investments over the years.

Ideally, after the user has ended (or abandoned) its current session on a logged-in device, so-called *de-authentication* must occur. However, de-authentication has received considerably less attention than authentication due to the perception that it is not as necessary as inadequate (or insufficient) authentication. Unfortunately, an unattended active secure session poses a real threat to *Lunchtime Attacks* [1]. Such attacks can occur whenever an adversary gains physical access to the active session of another user who carelessly stepped away and left the logged-in device unattended.

As a result, secure, privacy-preserving, and usable de-authentication techniques are needed. Prior results, however, do not satisfy all three requirements. Inactivity timeouts, for example, are a popular way of de-authenticating and can be considered somewhat[1] privacy-preserving. However, when timeouts are too long, security is poor as the lunchtime attack window expands. In contrast, if timeouts are too short, the user might have to re-authenticate needlessly [2]. Users are continuously authenticated by other methods, and once their identity can no longer be verified, they are de-authenticated. The most common technique relies on detecting a user's physical presence [3–5].

As a means of de-authentication, we believe continuous face recognition offers a promising solution. If the user's face is visible from the webcam, the system will track it and identify it; if the face is not visible (for a prescribed period of time), the system will de-authenticate them. This general approach offers several benefits. The first advantage is that it is simple to implement and requires no additional equipment since most modern general-purpose computers come equipped with webcams. Secondly, it is secure since current face detection algorithms are fast and highly accurate [6], making it resistant to *Lunchtime Attacks*. Finally, it keeps the user authenticated and logged in even if keyboard and mouse activities cease, provided they remain visible from the webcam. The method differs from inactivity intervals, keystroke dynamics [7] or gaze–tracking [1], which require users to interact with the system continuously or frequently.

However, de-authentication based on face recognition is hindered by significant **privacy concerns**, which should be carefully addressed [8]. To begin with, most users would not want to be continuously video recorded. The rules may explicitly state that recordings are not stored anywhere, but users might (rightfully) not trust such promises and refrain from (or attempt to circumvent) using such a method [9]. Additionally, the information captured by the webcam could also be exploited by an attacker who has gained access to it. An example of a possible threat would be blackmailing a user recorded during private moments.

Nonetheless, most modern devices have user-facing cameras, and despite the manufacturers' assurance that cameras only work when there is a visible indicator (e.g., an LED light next to, or in the camera itself), many users find the constant presence of the camera unsettling. In fact, on some computers with integrated cameras, it is possible to surreptitiously turn on the camera and record **without** triggering the obligatory indicator [10].

The safety and privacy concerns of webcams have led many cautious users to use physical barriers (e.g., placing tape) on their webcams [11]. Former FBI director James Comey [12] publicly supported this practice, and some laptop manufacturers now provide sliders to cover webcams.

Due to the discussion above, we propose BLUFADER, a continuous face detection and recognition de-authentication system that provides users privacy, security, and usability. We use a physical blurring material on the webcam that obfuscates the facial features of users and makes them unrecognizable. We demonstrate that our material is resistant to state-of-the-art de-blurring algorithms, and that privacy is preserved through two user studies. Following our demonstration that state-of-the-art face detection models fail to detect faces in blurred images, we developed a deep neural network to tackle the problem. Additionally, by leveraging a Siamese Neural Network, we enhanced BLUFADER security with a face recognition module that can distinguish users from their blurred images. We extensively tested our system with 30 subjects in different scenarios and activities, reaching over 95% accuracy. BLUFADER is an extension of the previous published approach BLUFADE [13]. In this extended version, we add the face recognition module to increase the overall reliability, assess the security of the system against de-blurring algorithms, and conduct user studies to evaluate the system's usability and users' perceptions.

Contributions:

- We propose a novel secure, usable, and privacy-preserving de-authentication method based on blurred face detection;
- We evaluate BLUFADER via extensive experiments, demonstrating that it outperforms state-of-the-art algorithms on blurred face detection tasks;
- We introduce a new continuous authentication mechanism based on blurred face recognition to enhance BLUFADER security against adversarial attacks;

[1] Timeouts are not very privacy-preserving since they monitor user's typing and/or mouse activity.

- We assess the security of BLUFADER against state-of-the-art de-blurring algorithms, validating the robustness of our method with a cohort of 34 participants.
- We assess the usability of BLUFADER by analyzing the impressions of a population composed by 27 individuals that tested the system for one hour.
- We publicly release two datasets of physically blurred faces: the first one consists of 20*k* images of celebrities and backgrounds, blurred with two different materials, and the second contains 1080 enrollment images and 600 videos of 30 subjects interacting with a laptop (both blurred).

Organization: Section 2 overviews related work. Section 3 describes the model, followed by Sections 4 and 5 which discuss the material evaluation and selection, respectively. Section 6 assesses the security of the selected filter against state-of-the-art deblurring algorithms, while Section 7 presents our experimental settings. Then, Sections 8 and 9 describe and report the results of the face detection and recognition modules of BLUFADER, respectively. Section 11 concludes the paper.

2. Related work

Related work stems from several areas, including continuous authentication and de-authentication (Section 2.1) as well as face recognition and detection models (Section 2.2).

2.1. Continuous authentication and de-authentication

In contrast with authentication techniques, which are extensively studied in the literature and are widely used in everyday life, there are no standard or broadly adopted user de-authentication methods. This reflects the fact that users are forced to authenticate at the beginning of a login session, while de-authentication is almost never mandatory. Locking the screen or logging out during a short break (e.g., coffee, bathroom, hallway chat, lunch) is widely perceived as being tedious or unnecessary (i.e., 25% of the users leave their computers unlocked when stepping away from their desk [14]). However, as mentioned earlier, failure to de-authenticate opens the door for lunchtime attacks, which are pretty common, as noted by Marques et al. [15]. Thus, the research community tried to come up with secure, usable, and privacy-preserving techniques for *automatic* user de-authentication.

The simplest de-authentication method is to log out the user after a fixed keyboard/mouse inactivity period. However, choosing the duration of this period is not trivial [2]. Recent techniques rely on Continuous Authentication (CAuth): the user is continuously monitored and authenticated while interacting with the system, and de-authentication happens once these interactions stop. CAuth usually relies on some form(s) of biometrics, usually based on the recognition of: face [3,16], voice [17], motion [18,19], keystroke and/or mouse dynamics [7], and even video-game playing style [20]. For an extensive list of these techniques, we refer to [21,22].

Of the above, keystroke dynamics is popular and seemingly non-intrusive while requiring no special equipment, whereas others need a camera and/or a microphone, which must be turned on. Keystroke dynamics utilize the user's unique typing style (reflected in a profile created at enrollment time) for authentication. While easy to deploy, this approach is not secure since an attacker can reproduce the user's typing style [23]. Carrying around a unique token that communicates with the workstation is another option [24]. However, its prominent drawback is the requirement to always carry and protect this token. A similar approach is explored in ZEBRA [4]: the user is continuously authenticated using a personal bracelet as long as wrist movements and the computer actions match. Unfortunately, [25] showed that Zebra is insecure. More complex and exotic systems, e.g., based on gaze–tracking [1] and pulse–response [26] have been proposed. Since they require pricey specialized equipment, thus their applicability is quite limited.

All aforementioned techniques have a major common drawback: a user can be authenticated only when **interacting** with the device. Consider the following frequent everyday activities that involve no interaction (no keyboard, mouse, or touchscreen actions) while the user remains physically present:

- Reading something on-screen or printed
- Watching a video/movie
- Listening to music or podcast
- Making a phone-call
- Taking a seated nap
- Having an in-person conversation with someone

Any of such activity, once it exceeds the inactivity threshold, would cause automatic de-authentication, resulting in extra user burden or even DoS. To overcome this issue, several methods have been proposed. FADEWICH [5] instruments an office with position sensors to detect whether the users are sitting at their desks. *Assentication* [27] detects user presence through pressure sensors in the chair cushion. Whereas, [28] instruments a chair with BLE beacons to detect whether the user is currently sitting. Facial recognition can be used for CAuth by continuously monitoring faces that appear in front of the camera, while being user-transparent [16,29,30]. In this paper, we focus mainly on the detection of faces, since most facial features would not be visible for privacy reasons, and tracking the face would be enough to assess the presence of a user. Then, we implement a face recognition algorithm to distinguish different users from their blurred images, although it cannot tell *who* they are.

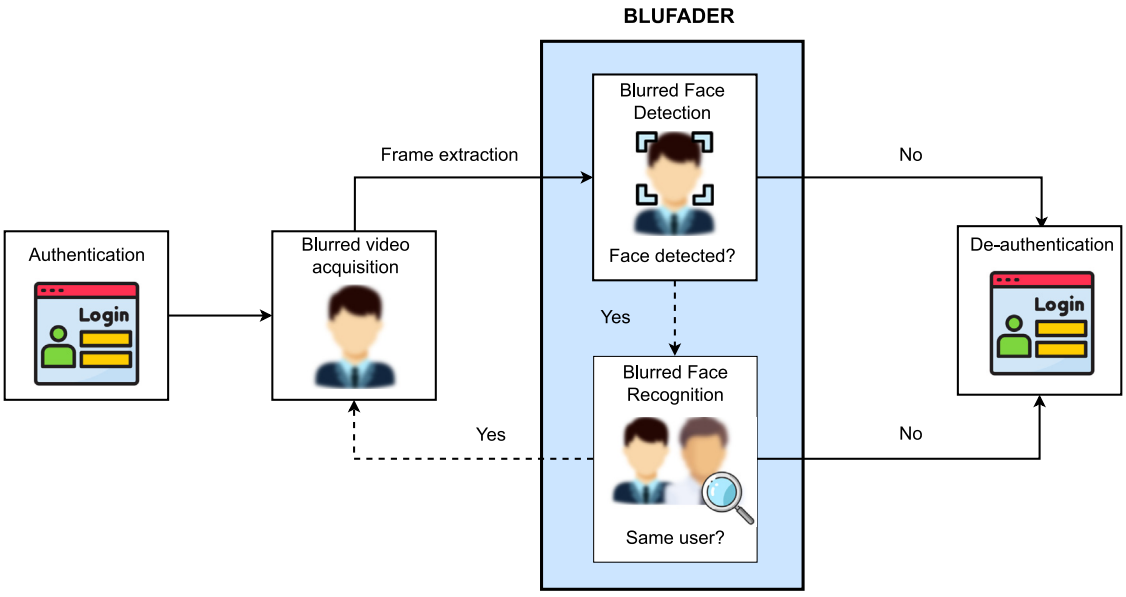


Fig. 1. BLUFADER operational flow.

2.2. Face detection and recognition

Face detection and face recognition are distinct Computer Vision tasks thoroughly studied in recent years. We consider face recognition a subclass of face detection, since the algorithms first start by detecting a face and then use its features to compare to a set of known faces to recognize the person. In the early stages, face recognition was done by automatically extracting distinctive facial features, e.g., eyes, mouth, or nose. These features were used to transform the face into a vector, and using statistical pattern recognition techniques, faces were matched [31,32]. With the rise of deep learning, especially Convolutional Neural Networks (CNN), computers reached (and surpassed) human performance in such tasks [33]. Deep-learning-based face recognition techniques can be divided into: (1) ones using single CNN [34,35], (2) multi CNNs [36], and (3) variants of CNN [37]. More recently, Siamese Neural Networks (SNN) are gaining traction in face recognition tasks [38,39], given their ability to scale for many classes (i.e., different people) with just a few samples each. For a comprehensive list of face recognition methods, refer to [40,41].

Similar to face recognition, early face detection methods were based on developing discriminative hand-crafted features from faces and building robust learning algorithms [42,43]. Nowadays, with the evolution of CNNs, detecting frontal faces is considered a solved task [6]. More efforts took place to detect faces under challenging conditions, such as partial faces [44] or faces captured by depth sensors [45]. Recently, TinaFace [46], by considering face detection as a particular object detection task, outperformed state-of-the-art methods on the set of most challenging face detection dataset WIDER FACE [47]. We refer to [48] for a complete treatment of this topic. Finally, [49] tested state-of-the-art face detection models on low-quality images with different levels of blurring, noise, and contrast, showing that both hand-crafted and deep-learning-based face detectors perform poorly on such images.

3. Model overview

In Fig. 1, we show the operational flow of our approach. At the beginning of the session, the user logs in through canonical methods, such as providing a password or through fingerprint recognition (Authentication). Once authenticated, a physically blurred webcam starts recording (Blurred Video Acquisition). Then, BLUFADER keeps the user logged in by continuously detecting (Blurred Face Detection) and recognizing (Blurred Face Recognition) the user's face. If any of these conditions are not met BLUFADER terminates the session by requesting login credentials from the user again. In this section, we describe in detail our system model and its real-world application scenarios.

3.1. Threat model

BLUFADER purpose is two-fold: (i) protect users against the so-called *lunchtime attack* [1,28] through de-authentication, (ii) solve the privacy issues related to face recognition authentication algorithms. Therefore, we identify two kinds of attackers in our threat model. The *Attacker I* wants to gain access to the logged-in session of the victim when they step away from their workstation. Attackers and victims might know each other or be unrelated, as long as they can

both physically reach the victim's workstation. For instance, they could be both employees working from a shared office. *Attacker II* wants to violate the user's privacy by accessing and/or elaborating the video stream of the webcam used for face recognition. This attacker can be the (untrusted) developer of the face recognition system, or malware previously installed.

Formal definition. We now describe the scenario where the attackers operate, according to the following four criteria [50]: goal, knowledge, capability, and strategy. For clarity, we divide the two attackers.

3.1.1. Attacker I (lunchtime attack)

- *Goal:* The attacker ultimately wants to take control over a logged-in user session of a target victim. Once the access is obtained, the attacker can conduct a plethora of attacks (e.g., installing a backdoor, deleting data) which are out of this paper's scope.
- *Knowledge:* The attacker does not know the victim's credentials, nor any other information to access the victim's user session legitimately. The only information available to the attacker is when the workstation is left unattended with a logged-in session. This information can be obtained by directly looking at the workstation, or through indirect means (e.g., a remote camera, an informer accomplice).
- *Capability:* The attacker can monitor the victims to spot when their workstation is left unattended. The attacker might have accomplices to distract and create a pretext to let the victim leave their computer unattended.
- *Strategy:* The attacker waits for the right moment (i.e., opportunistically) when the victim leaves the (logged-in) machine unattended, or pushes, possibly with collaborators, the victim to do so (i.e., orchestrate the attack). As soon as the victim loses eye contact with their machine, the adversary takeover the terminal.

We assume the attackers might be motivated by many reasons. For instance, they can be co-workers aiming for the same position, or the victim could have high-value secrets stored in their machines. Even a short exposure of the unlocked computer to the attacker may result in dangerous consequences such as installing malware.

Practical scenario. Our threat model comprises three stages an attacker must accomplish: *wait, approach, and leave*.

1. *Wait:* The adversary monitors the victim (physically or remotely), waiting until the victim steps away from the workstation while still logged in.
2. *Approach:* At this point, the attacker takes control of the machine by leveraging the active session of the victim.
3. *Leave:* Finally, the attacker leaves the computer before the victim comes back without leaving traces. An accomplice may help the attacker to leave in time by signaling the imminent return in advance.

The *wait* phase includes a period in which the victim would be able to spot the attackers taking over. In the literature, this period is called *grace period*, and in previous approaches has been set to 6 s [5]. Therefore, any de-authentication methods occurring within 6 s are considered secure and reliable. To deploy BLUFADER, our primary requirement is to keep the grace period as short as possible, restricting the attack window.

3.1.2. Attacker II (privacy violation)

- *Goal:* The attacker wants to violate the victim's privacy by accessing the video stream when using a face recognition system (e.g., recording private moments of the victim and blackmailing them).
- *Knowledge:* The attacker does not require any particular knowledge about the victim.
- *Capability:* We assume the attacker has access to the video stream of the victim's webcam. This might happen when the camera is not used on purpose (i.e., should be off), during on-purpose usage (e.g., video conferencing), or when a face recognition system is running (the webcam should record the user only for (de-)authentication purpose).
- *Strategy:* The attacker continuously records the video stream of the victim and inspects it afterward to spot private segments.

We assume the attackers might be motivated by many reasons, such as monetary purposes (e.g., blackmailing), besmirching reputation, or stalking.

Practical scenario. Our threat model comprises three stages an attacker must accomplish: *access, record, and inspect*.

1. *Access:* The adversary gains access to the victim's webcam. How to accomplish this goal is out of this paper's scope (see [10], for instance).
2. *Record:* The attacker continuously records the video stream from the victim's webcam.
3. *Inspect:* Finally, the attacker inspects the recording and extracts the victim's private data.

In this scenario, the attacker might be the manufacturer of the face recognition system (often untrusted by users [11]), or any party interested in violating the user's privacy (e.g., a stalker, an employer to monitor an employee). We remark that BLUFADER aims to protect the users' privacy in all situations where the camera is not used on purpose, and not to block access to the webcam by an attacker (which is out of our scope). Indeed, if the user wants to make themselves visible (e.g., for a video conference), BLUFADER cannot protect the user's privacy. On the contrary, if the webcam is not supposed to be active, or should record the user only for secondary purposes (e.g., for continuous authentication), BLUFADER can effectively preserve privacy. The latter scenario is the main one we focus on and contribute to: preserving

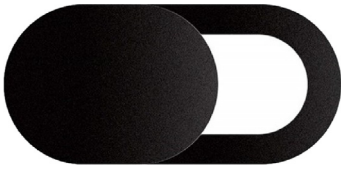


Fig. 2. Webcam slider cover example.

users' privacy while using a face recognition system. In particular, the user's presence might still be inferred (otherwise, the authentication method could not work), but BLUFADER should effectively hide all contextual information that can compromise the user (e.g., facial expressions, the surrounding environment, actions taken, lip movements). Last, our threat model does not directly consider background users, since the high blur level should intrinsically make them undetectable. More studies will be conducted in future works to overcome this limitation.

3.2. System model

The core idea is to use a webcam (built-in or external) to detect the user's face continuously. At the beginning of the session, the user authenticates by any canonical method, e.g., passwords or fingerprint recognition. Then, BLUFADER collects images at regular intervals from the webcam, keeping the user authenticated as long as a face is detected. Once the detection fails and a grace period passes, the user is automatically logged out. To preserve user privacy, the webcam view is *physically* blurred by a somewhat transparent tape or a similar means. Thus, users can be sure that the images received by the webcam are already altered and cannot be used to recognize them. We note that BLUFADER's goal is to detect, and not to actually recognize, faces since the tape should blur the image enough to obscure facial traits. Indeed, we designed our recognition model to distinguish users from their blurred images, but it cannot tell the users' identity.

Besides privacy, BLUFADER offers the usual benefits of face detection de-authentication mechanisms. First, is completely transparent for the user, since it does not interfere with normal user behavior, and prevents *Lunch Time Attacks*. Furthermore, it only requires a simple strip of tape as additional equipment, and allows the user to remain inactive without being de-authenticated, as long as they remain in the camera's view. The main implementation challenges are: (i) selecting an appropriate material that obscures users' facial traits, while still allowing face detection by automated algorithms, (ii) developing an algorithm to detect faces from blurred images, and (iii) implementing a model to perform facial recognition from blurred images. (i) is analyzed in Section 5, (ii) in Section 7, and (iii) in Section 9.

3.3. Daily usability

In everyday use, it may be impractical to have a physical material such as tape applied in front of the webcam. The material should be removed and reapplied every time the user wants to join a video call with a webcam enabled or, in general, use the camera for any other activity. In the long run, this can eventually damage the device. To address such a problem, a slider cover can be applied on top of the webcam to switch between blurred and clean webcam view easily. In Fig. 2, a standard slider cover is shown. Our slider cover would include the blurring material to cover the webcam.

Subverting the system. A sophisticated ad-hoc attack against BLUFADER would see the attacker impersonating the victim via their printed face or wearing a mask. We argue this scenario would be challenging to achieve for two reasons: firstly, the grace period complicates the feasibility of this approach. In fact, in less than 6 s, the attacker must be fast enough to cover up his face and take the victim's place, without being noticed. Our grace period (far below 6 s) has been widely discussed in Section 8.4, making this threat even more unrealistic. Secondly, the adversary must consider possible people who will notice the suspicious behaviors. Due to these reasons and the resulting small likelihood of success, we excluded this attack from the threat model.

3.4. Application scenario

We start by distinguishing between shared and personal computers. We assume that the latter is always used by the same person; thus, the detection system can be tailored to their blurred face. The phase of training the software to recognize a face is called *enrollment*. In shared computer settings, the system is used by multiple users and should detect all of them. Thus, the enrollment is complicated and should either be done for every new user or require having their enrollment images a priori, which is clearly not applicable. The second distinction concerns the place where the system is used. A computer can be stationary or portable, which defines the scene its webcam sees when no users are present (i.e., the "background"). If stationary, the background is fixed; otherwise, it will vary depending on the place. Based on that, we identify four scenarios:

- **Scenario 1 - Same person and fixed background:** represents workstations or desktops, located in an office/home and is always used by the same person. Enrollment is possible;

- **Scenario 2 - Different people and fixed background:** represents shared workstations in fixed places (e.g., offices). Enrollment is not applicable;
- **Scenario 3 - Same person and variable background:** represents personal computers, e.g., laptops or tablets, that owners can bring anywhere. Enrollment is possible;
- **Scenario 4 - Different people and variable background:** represents shared computers that are either portable and/or have variable backgrounds, e.g., public ATMs or wheeled workstations. Enrollment is not applicable.

4. Material evaluation

One of the critical design elements for BLUFADER is how to choose the appropriate blurring material. In this section, we discuss the criteria for this selection (Section 4.1), and the experimental settings to determine the best candidates in terms of suitability for face detection (Section 4.2).

4.1. Selection criteria

The ideal blurring material should satisfy three requirements: (i) blur enough to prevent face recognition, (ii) not blur too much to enable face detection, (iii) be inexpensive and readily available. Based on these requirements, we identify five possibilities[2]:

- ***Chair*** - Polimark Poliver Battisedia 280 854. Semi-transparent rigid plastic material that is commonly used on floors to prevent chairs from scratching them;
- ***Antirefl*** - Polimark Poliver PL01322. Anti-reflective obfuscating film, commonly used on windows to block visibility from the outside but let light pass through;
- ***Ruvid*** - Ruvid Transparent Paper. Transparent rough paper used as book covers;
- ***RuvidX2*** - Double Ruvid Transparent Paper. Double layer of the previous item;
- ***Scotch*** - Magic Tape Scotch 3M. Common semi-transparent white adhesive tape;

4.2. Experimental settings & best candidates

To find the best blurring material, we evaluated the quality of blurred images produced by a webcam when various materials were applied. To this extent, we used a mannequin called Dolores[3] as a fixed subject of our photos. For each material, we positioned Dolores in front of the webcam at several distances (from 30 cm to 90 cm, with 10 cm steps), simulating realistic usage scenarios. We used a white background in a light-controlled environment. At each distance, we took five snapshots, and used three samples of each material. Then, we assessed image quality (i.e., sharpness) using the algorithm presented in [51], and averaged the results. Fig. 3 shows pictures of Dolores taken with different blurring materials, while Fig. 4 shows the quality of images for all materials and steps. A lower Niqe value indicates the image has a higher sharpness. The plot shows that all blurring materials significantly lower image quality and that the distance from the webcam does not meaningfully influence the Niqe value. Ideally, the lower the image quality, the more challenging the face recognition by automatic systems. Thus, we selected two materials yielding the highest quality images (*Chair* and *Antirefl*), which from visual inspection (examples are visible on Fig. 3), could preserve users' privacy. The following section provides more evidence of their privacy features and discusses material selection.

5. Material selection

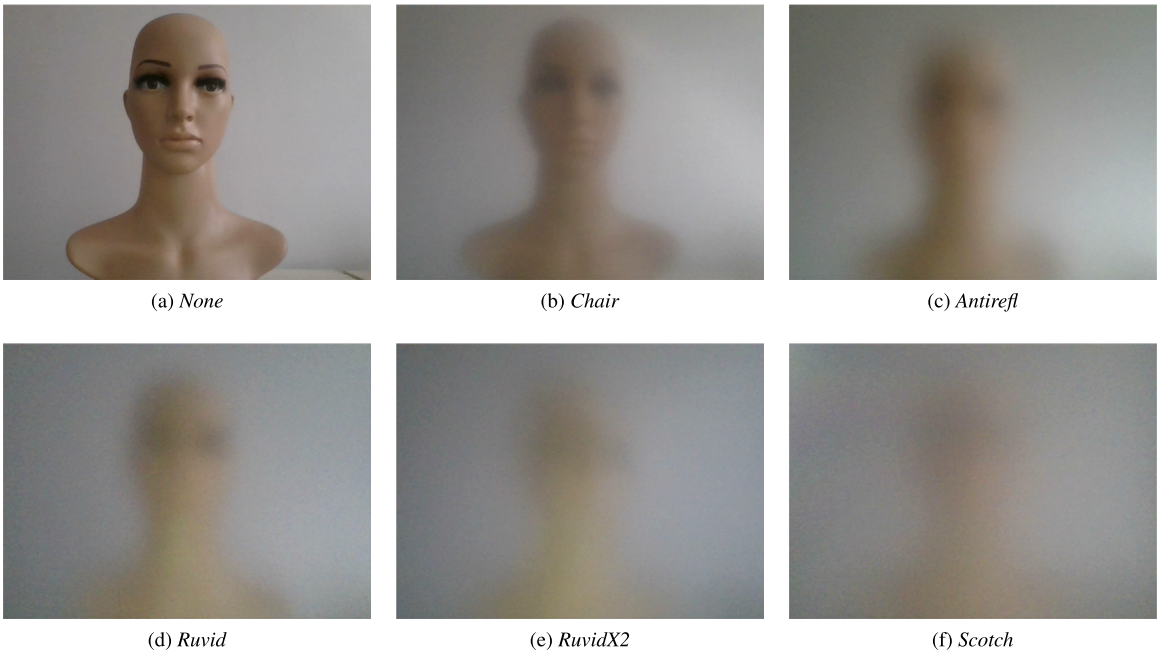
To select the best material among the two candidates from the previous section, we must evaluate their privacy-preserving characteristics. To this extent, we first collected a dataset of blurred pictures of celebrities (Section 5.1), and we conducted a survey asking the participants to recognize some of them (Section 5.2). Last, we report the results and final decision (Section 5.3).

5.1. Celebrities dataset

To the best of our knowledge, there are no physically blurred faces datasets publicly available. Furthermore, to carry on our experiments, we need images of both blurred backgrounds and faces with the materials we selected in Section 4.2. To create such a dataset, we exploit the CelebA dataset [52] and the SUN dataset [53]. In particular, we randomly selected 5000 images from CelebA (faces) and 5000 images from SUN (backgrounds). Then, applying the *Chair* and *Antirefl* filters to a laptop webcam, we recorded a slideshow of the 10K images displayed on a tablet. Finally, we picked a frame in correspondence of each image from the recording, creating two new datasets of 10K blurred images each. The dataset is available at the following link: https://spritz.math.unipd.it/projects/BLUFADE/

[2] Chair: https://bit.ly/3i9Vjm8, Antirefl: https://bit.ly/3CN14xS, Ruvid: https://bit.ly/3m3KZ0i, Scotch: https://bit.ly/3zMUOV8.

[3] The name was chosen from an analog situation from the TV series Umbrella Academy.



(a) *None* (b) *Chair* (c) *Antirefl*

(d) *Ruvid* (e) *RuvidX2* (f) *Scotch*

Fig. 3. Effectiveness of blurring materials considered at a distance of 30 cm.

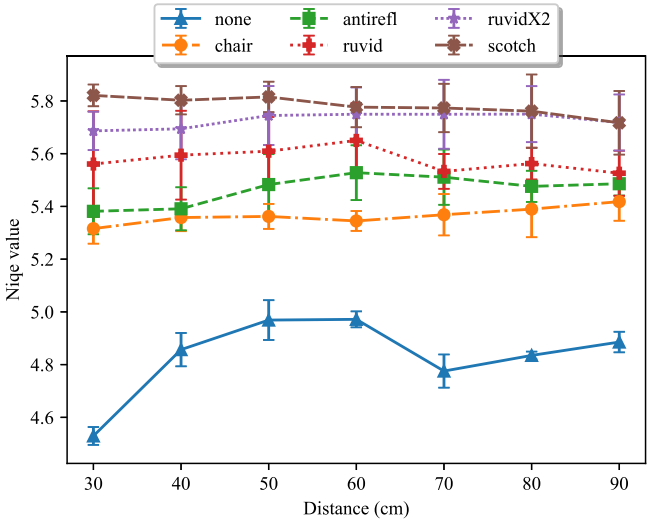
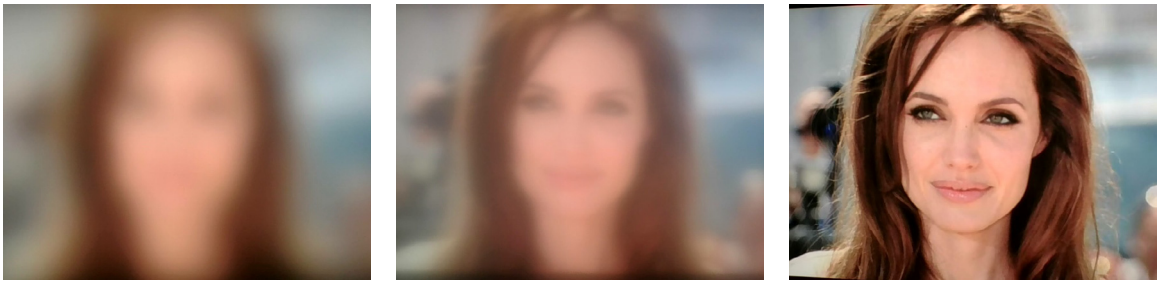


Fig. 4. Averaged quality of images for each material and steps. Lower Niqe values are associated to sharper images.

5.2. Celebrities privacy survey

We conducted an online survey[4] asking participants to recognize celebrities from blurred images to test whether the blur level was enough to protect users' privacy. In particular, we selected ten images of well-known celebrities in a neutral context, and we asked participants to guess their names. For each image, first, we presented the *Antirefl* version, then the *Chair* version, and last the original image (i.e., from the less sharp image to the most). The participants were asked to provide a name at each step, without the possibility of going back and changing the name after seeing a less blurred image. If the name provided at the last step was correct (we also accepted names with spelling errors), we could assume the participant knew the celebrity, and thus we checked at which blur stage the participant recognized them. If the participant

[4] https://forms.gle/soMtKSLz9z8h5LFr7.



(a) *Antirefl* (b) *Chair* (c) *None*

Fig. 5. Angelina Jolie with different blur filters.

Table 1
Average *NIQE* score for images with no filter applied (Sharp), *Antirefl* filter, and the three deblurring models applied on the *Antirefl* images.

	Sharp	*Antirefl*	*MIMO-UnetPlus*	*Maxim*	*DeepRFT*
NIQE	3.578 ± 0.522	3.893 ± 0.606	4.628 ± 0.536	3.838 ± 0.634	3.938 ± 0.847

did not know the celebrity, we discarded that sample. Fig. 5 shows an example of a celebrity blurred with the two filters and the original photo. In Appendix A we report one of the set of questions related to a single celebrity.

5.3. Survey results and material decision

We collected answers from 70 participants (Age range: 22–45, 64.3% Male, 35.7% Female). 391 images were recognized correctly with no blur, 273 with *Chair* blur, and only 5 with *Antirefl*. In other words, participants recognized a celebrity they knew only in 1.28% of the cases through the *Antirefl* filter, and in 69.8% of the cases through *Chair*. Thus, we demonstrated that *Antirefl* successfully protects users' privacy, and we decided to use it for the rest of the experiments.

6. Material blurring assessment to adversarial attacks

In Section 5, we demonstrated the effectiveness of *Antirefl* as a physical blurring filter. However, many methodologies have been successfully applied to video and image deblurring tasks [54], highlighting the need for a robustness assessment of our physical filter to such approaches. We selected state-of-the-art models for image deblurring and applied them to our Celebrity dataset blurred using the *Antirefl* filter. Further, we assessed the efficacy of the deblurring algorithms through a survey[5] structured as the one presented in 5.2.

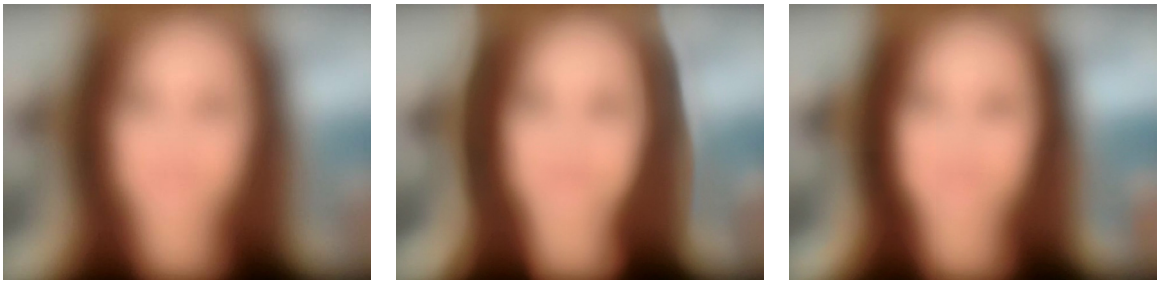
6.1. State of the art deblurring algorithms

To the best of our knowledge, there are no deblurring models specifically designed for reconstructing physically blurred images. In this context, we selected three state-of-the-art deblurring pre-trained models (i.e., MIMO-UNet [55], Maxim [56] and DeepRFT [57]) that showed the best performances[6] on the GoPro dataset [58].

The GoPro dataset comprises real-world blurred frames and ground truth sharp images from an acquisition system combined with postprocessing techniques. We process the same 5000 cluster of samples of celebrities for the three selected models, using the *Antirefl* filtered version of images. Improvement in sharpness (i.e., NIQE) is reported in Table 1. Results show that the sharper images are obtained from the *Maxim* model, while other algorithms tend to degrade the image's sharpness slightly. To prevent any doubt of performances regarding Maxim and DeepRFT models (i.e., the second deblurring method after Maxim based on Table 1), we also carefully verified the results obtained with an *unpaired t test*. The result shows that the two-tailed P value is less than 0.0001, therefore we can consider the difference between the two models to be statistically significant. In Fig. 6 we show an example in which we compare the images reconstructed by the deblurring algorithms applied on *Antirefl* filter.

5 https://forms.gle/DsaQt5aG774ye3XF6.

6 https://paperswithcode.com/sota/deblurring-on-gopro.



(a) *MIMO-UNet* (b) *Maxim* (c) *DeepRFT*

Fig. 6. Deblurring algorithms output on *Antirefl* Angelina Jolie blurred image.

6.2. Deblurred celebrities privacy survey

To validate the results that we got from the deblurring models, we proposed an additional survey. We asked participants to recognize, well-known celebrities in a few photo versions of the same person. The images selected were all different from those included in the questionnaire presented in Section 5.2. We submitted 10 famous individuals, proposing three different versions of the same image: *Maxim* output image, the original *Antirefl* blurred image and the sharp original one. Appendix B shows questions related to one of the 10 celebrities involved in the survey. The sample was discarded if the participant could not recognize the clear image. At the same time, the rest of the answers were valuable to measure the effectiveness of the deblurring processing. The survey reached out 34 participants (79,4% of males and 20,6% females, age ranging from 22 to 49 years). The results highlight the evident difficulty of recognizing celebrities despite the *Maxim* deblur applied. Out of 148 correctly recognized famous characters in clear images, only 4 samples were correctly identified with the original *Antirefl* blurring. Interestingly, the most crucial insight is that only 3 characters have been discovered with the *Maxim* deblur applied, demonstrating the robustness of *Antirefl* to state-of-the-art deblurring algorithms.

7. Experimental setting

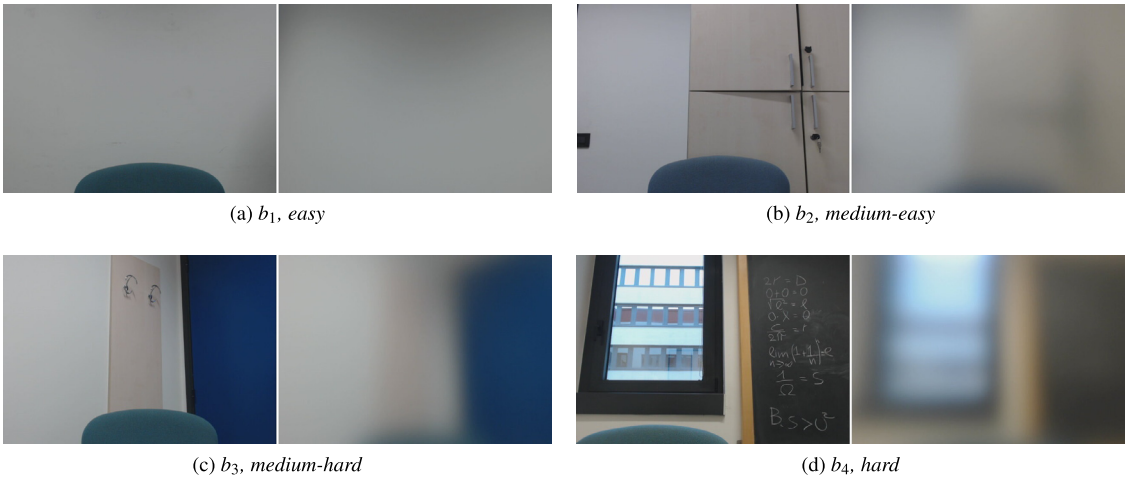
We now present the experimental settings we adopted to evaluate BLUFADER. In Section 7.1, we illustrate the data we collected for the experiments. Section 7.2 presents how we designed the four scenarios to evaluate our systems. The experimental settings explained in this section will be adopted both in face detection and recognition models.

7.1. Data collection

To conduct our experiments, we collected data from 30 people, 13 females and 17 males, aged 22–43. According to the scenarios presented in Section 3.4, we first asked participants to follow an enrollment procedure, then recorded them while performing common everyday actions. In detail, the enrollment procedure consisted in taking snapshots of the user in 9 different positions: in front of the webcam at a close distance (i.e., less than 30 cm), mid-range distance (between 30 and 70 cm), and far (more than 70 cm); at mid-range translated to left and right (i.e., the face should be completely contained in the left or right half of the webcam view); at mid-range rotating the head by looking up, down, left, and right. Then we recorded users for 10 s while reading an email, writing sentences, looking at their phones, talking with a colleague, and leaving the workstation. Users repeated these steps on four different *backgrounds* $b_n \in \mathcal{B}, n = \{1, 2, 3, 4\}$ of increasing difficulty: a white wall (b_1 - easy), a white wall with a closet and a poster (b_2 - medium–easy), a white wall with a blue door (b_3 - medium–hard), a white wall with a written blackboard and a window (b_4 - hard). They are shown in Fig. 7. We used a Logitech C922 Pro Stream Webcam (30 Frames Per Second) with *Antirefl* blur for the recordings. This dataset is available at the following link: https://spritz.math.unipd.it/projects/BLUFADE/

7.2. Four scenarios

To represent the four scenarios from Section 3.4, we used the enrollment snapshots and the activity videos to create different training and test sets. In general, enrollment images are used in the training set, while activity videos are used for testing. For each scenario, we test person by person and background by background, creating every time a training set that respects the requirements of the scenario to fine-tune the neural network. We remind that every person p of our dataset of people $\mathcal{P}$ has taken 9 enrollment snapshots for each background b of the 4 backgrounds $\mathcal{B}$ analyzed (from easy to hard). We refer to the 9 enrollment images of a person p in a background b as $e_{p,b}$. In more detail, we use a leave-one-out procedure, testing at each iteration the activity videos of a person $p \in \mathcal{P}$ in a background $b \in \mathcal{B}$, and setting the training (fine-tuning) set as specified in Table 2. In the table, we give formal and informal explanations on how we constructed the training set to understand the scenarios easily.



(a) b_1, *easy* (b) b_2, *medium-easy*

(c) b_3, *medium-hard* (d) b_4, *hard*

Fig. 7. The four different backgrounds used in the experiments (left original, right blurred with *Antirefl*).

Table 2
Training set composition according to specific application scenario. $\mathcal{P}$ and $\mathcal{B}$ are sets of participants and backgrounds, respectively.

Scenario	Formal training set	Explanation
(1) Same person and fixed background	With p, b fixed, $i \in \mathcal{P}$, $j \in \mathcal{B}$, $\bigcup_{\substack{\forall i \neq p \\ \forall j \neq b}} e_{i,j} \cup e_{p,b}$	All enrollment snapshots of people different from p in backgrounds different from b + enrollment of p in background b
(2) Different people and fixed background	With p, b fixed, $i \in \mathcal{P}$, $j \in \mathcal{B}$, $\bigcup_{\substack{\forall i \neq p \\ \forall j}} e_{i,j}$	All enrollment snapshots of people different from p
(3) Same person and variable background	With p, b fixed, $i \in \mathcal{P}$, $j, k \in \mathcal{B}$, $\bigcup_{\substack{\forall i \neq p \\ \forall j \neq b}} e_{i,j} \cup e_{p,k} \mid k \neq j$	All enrollment snapshots of people different from p in two backgrounds (j, k) different from b + enrollment of p in the remaining background different from b, j, k
(4) Different people and variable background	With p, b fixed, $i \in \mathcal{P}$, $j \in \mathcal{B}$, $\bigcup_{\substack{\forall i \neq p \\ \forall j \neq b}} e_{i,j}$	All enrollment snapshots of people different from p in backgrounds different from b

7.2.1. Regular people vs. Celebrities

To introduce more variance in the training set, we also run some experiments using the celebrities dataset. The first set of experiments was run using only people's snapshots as a training set. Then, we repeated the experiments by adding in training set 1080 celebrities' faces, and the last repetition was done by fine-tuning the network using celebrities only. This way, we could see how the variance in the training set affects the performance of network detection. The case of celebrities only was possible just in the fourth scenario, since it was impossible to have their enrollment or more celebrities in the same background.

8. Face detection

The first component of BLUFADER consists of the face detector. The purpose of this module is to detect the presence of a face in a blurred image. As long as the system maintains face tracking, the user remains authenticated; otherwise, it terminates the session, de-authenticating the user.

In this section, we first discuss state-of-the-art performance in blurring face detection tasks (Section 8.1). Then, in Section 8.2 we introduce our approach for blurred face detection. Further, in Section 8.3, we show the performance of our model. Finally, in Section 8.4, we evaluate the performance of BLUFADER in de-authenticating people.

8.1. State of the art face detection algorithms

The performance of BLUFADER highly depends on the face detection algorithm behind it. Before implementing our neural network, we tested the state-of-the-art face detection systems on both our celebrities and enrollment blurred images of our participants. To this extent, we extracted 240 random celebrities and 240 random enrollment images and tested with Google Cloud Vision,[7] Amazon Rekognition,[8] Azure Cognitive Services with detection_01 and detection_03

[7] https://cloud.google.com/vision/docs/detecting-faces.

[8] https://docs.aws.amazon.com/rekognition/latest/dg/faces.html.

Table 3
Comparison between accuracy of state-of-the-art face detection models on blurred samples from Celebrities and People datasets.

	Google	Amazon	Azure v1	Azure v3	TinaFace
Celebrities	1.67%	43.75%	0.04%	45.83%	13.75%
People	3.33%	26.25%	0.00%	72.08%	18.75%

models,[9] and TinaFace [46]. Results are reported in Table 3, and they show how any of the state-of-the-art models were not suitable for our task, given the high blur level of our images. Even Azure v3, explicitly designed for blurred faces, with 72.08% of accuracy, was not good enough for BLUFADER.

8.2. Proposed face detection model

The poor performances of state-of-the-art methods in detecting blurred faces suggest that a new approach is needed for this task. Since the high level of blur removes facial traits, we decided to shape our problem as an object detection task, as also suggested by Zhu et al. [46]. Rather than binary classification (i.e., face vs. no face), we opted for object detection to possibly track the person, or detect two or more people in the same image for security purposes. For instance, if a person is logged in and using the computer and another user walks behind the first user, the system should detect which person is keeping the session alive; otherwise, it might wrongly de-authenticate the user. Furthermore, [49,59] demonstrated that CNNs do not cope well with blurred images, but fine-tuning them can help to improve the performances in object detection significantly. From these considerations, we decided to fine-tune the state-of-the-art object detection model RetinaNet [60], which uses ResNet and Feature Pyramid Network as the backbone for feature extraction. We followed an official procedure released by TensorFlow [61]. In particular, our fine-tuning procedure follows these steps: starting from ResNet pre-trained using the COCO dataset [62], we replace the classification head with a new randomly initialized classification head able to classify a single class (i.e., face), and we finally fine-tune the network using 150 batches of 32 samples each, with SGD optimizer (learning rate = 0.01, momentum = 0.9).

8.2.1. Confidence threshold

RetinaNet returns the objects it detects along with their confidence scores. Based on a threshold, usually 0.80, the object is detected or ignored. Since our data is highly blurred and strongly differs from usual data, we had to find a proper threshold for the task. We used the more general celebrities in this case since it has thousands of faces and thousands of backgrounds without faces. Using the celebrities instead of the people dataset to find the threshold, we would have limited the possibility of overfitting. Thus, we fine-tuned the network with the same 1080 celebrities we used to augment the people training set, and we tested the network on the remaining celebrities and backgrounds of our dataset. We tested thresholds ranging in [0.100, 0.125 0.150, . . . , 0.900]. The detection accuracy per threshold is presented in Fig. 8. We reached the best accuracy (82%) with the threshold = 0.425, which we used for the rest of the experiments. We remind the reader that an object detection threshold is significantly different from a binary classification threshold. In a binary classification task, the threshold defines whether the prediction goes to one class or another. In an object detection task, confidence defines the probability of an object being present (0 certainly not present, 1 certainly present). Setting a low confidence threshold for a class means increasing the recall of detecting that class despite (usually) lowering the precision. Setting a proper threshold (e.g., by looking at a ROC Curve) is a common approach to increase the model's performance according to the use case. For instance, in autonomous driving, the class "people" is purposely detected with low confidence (i.e., leading to a high recall) to avoid any critical false negative [63](e.g., the person is not detected and possibly invested).

8.3. Face detection performance

Table 4 reports the balanced accuracy of face detection on the frames of the activity videos divided by scenarios, backgrounds, training datasets, and tasks (T1 = read email, T2 = write sentence, T3 = look phone, T4 = talk with colleague, T5 = leave workstation). As expected, we reach the best performance on the easiest background $b1$, with around 98% accuracy on every scenario using the people scenario, 97% also using the celebrities, and 94% in the celebrities-only case. Among the tasks, T1, T2, and T3 scores the best, probably because are composed of the frontal frames of the people. In T4, people were talking with a colleague on their left or right, showing the webcam their face profile. This has probably led to some mistakes. Finally, T5 shows some errors during the transition period in which the user is leaving. In fact, we considered the user had entirely left only when the face was not more visible, and the network struggled a bit with partial faces or with just the body. However, when the user was fully present or absent, the network worked just fine, as in the other tasks. In Section 8.4, we better analyze this task to implement the de-authentication system.

[9] https://docs.microsoft.com/en-us/azure/cognitive-services/face/face-api-how-to-topics/specify-detection-model.

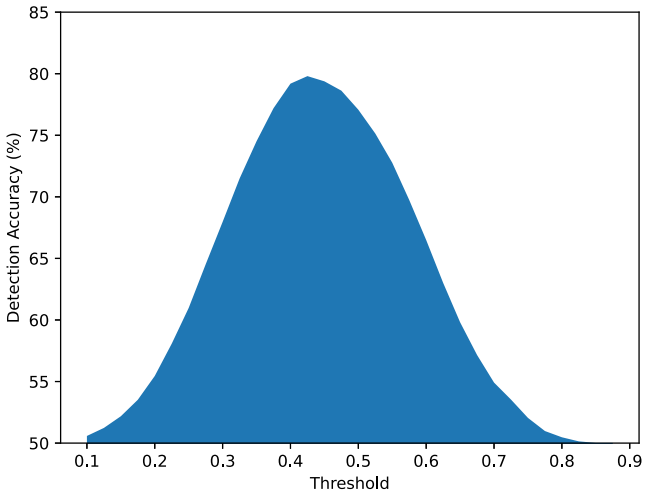
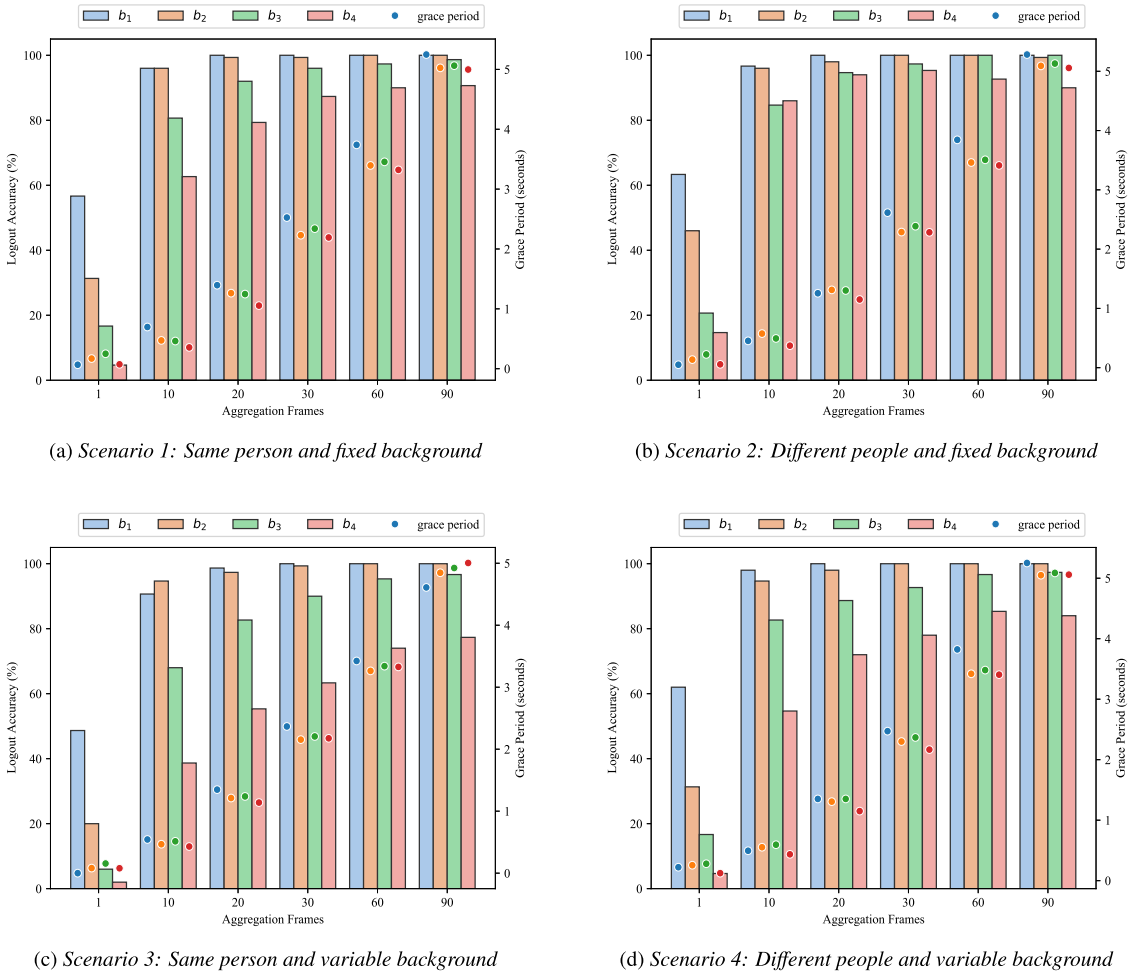


Fig. 8. Detection accuracy trend against the threshold.

Table 4
Balanced accuracy of face detection of frames of activity videos divided by scenarios, backgrounds, training datasets, and tasks (T1 = read email, T2 = write sentence, T3 = look at phone, T4 = talk with colleague, T5 = leave workstation).

Training	Task	Scenario 1					Scenario 2					Scenario 3					Scenario 4				
		b_1	b_2	b_3	b_4	**Avg**	b_1	b_2	b_3	b_4	**Avg**	b_1	b_2	b_3	b_4	**Avg**	b_1	b_2	b_3	b_4	**Avg**
People	T1	1.00	0.99	0.95	0.86	**0.95**	1.00	0.99	0.95	0.91	**0.96**	0.99	0.99	0.93	0.80	**0.93**	0.99	0.98	0.93	0.82	**0.93**
	T2	1.00	0.99	0.95	0.87	**0.95**	1.00	0.99	0.95	0.94	**0.97**	0.99	0.99	0.94	0.80	**0.93**	1.00	0.99	0.93	0.83	**0.94**
	T3	0.99	0.99	0.90	0.84	**0.93**	1.00	0.99	0.92	0.93	**0.96**	0.99	0.99	0.89	0.78	**0.91**	0.99	0.98	0.88	0.79	**0.91**
	T4	0.98	0.94	0.88	0.80	**0.90**	0.98	0.96	0.90	0.93	**0.94**	0.98	0.95	0.87	0.73	**0.88**	0.99	0.94	0.86	0.72	**0.88**
	T5	0.94	0.94	0.91	0.77	**0.89**	0.94	0.94	0.92	0.79	**0.89**	0.91	0.90	0.88	0.68	**0.84**	0.94	0.93	0.90	0.75	**0.88**
	Overall	**0.98**	**0.97**	**0.92**	**0.83**	**0.92**	**0.98**	**0.98**	**0.93**	**0.90**	**0.95**	**0.97**	**0.97**	**0.90**	**0.76**	**0.90**	**0.98**	**0.97**	**0.90**	**0.78**	**0.91**
People & Celeb	T1	0.99	0.98	0.94	0.74	**0.91**	0.99	0.99	0.96	0.90	**0.96**	0.99	0.99	0.93	0.72	**0.91**	0.99	0.99	0.94	0.78	**0.93**
	T2	0.99	0.98	0.95	0.80	**0.93**	0.99	0.99	0.96	0.93	**0.97**	0.99	0.99	0.94	0.80	**0.93**	0.99	0.98	0.96	0.83	**0.94**
	T3	0.99	0.98	0.87	0.72	**0.89**	0.99	0.99	0.90	0.88	**0.94**	0.99	0.98	0.84	0.68	**0.87**	0.99	0.98	0.87	0.74	**0.90**
	T4	0.94	0.89	0.84	0.62	**0.82**	0.95	0.91	0.84	0.82	**0.88**	0.94	0.90	0.79	0.61	**0.81**	0.95	0.90	0.82	0.66	**0.84**
	T5	0.93	0.92	0.90	0.74	**0.87**	0.94	0.92	0.91	0.78	**0.89**	0.94	0.92	0.88	0.73	**0.87**	0.93	0.91	0.89	0.73	**0.87**
	Overall	**0.97**	**0.95**	**0.90**	**0.72**	**0.89**	**0.97**	**0.96**	**0.91**	**0.86**	**0.93**	**0.97**	**0.96**	**0.88**	**0.71**	**0.88**	**0.97**	**0.95**	**0.90**	**0.75**	**0.89**
Celeb	T1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.99	0.95	0.91	0.65	**0.87**
	T2	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.99	0.97	0.93	0.73	**0.91**
	T3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.96	0.94	0.76	0.57	**0.81**
	T4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.84	0.79	0.73	0.52	**0.72**
	T5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	0.92	0.88	0.88	0.70	**0.84**
	Overall	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	**0.94**	**0.91**	**0.84**	**0.63**	**0.83**

Looking at the scenarios, surprisingly, those without enrollment (i.e., Scenarios 2 and 4 that consider different people) show slightly better performances than the others. This could be explained by the lacking of real unique traits in the enrollment images. Having a wider variance in the training set helps the network in detecting people in different tasks. In fact, the significant differences are again in T4, and T5, thus a more general network can help in such complex tasks. Finally, better performances are achieved when the training set is formed by people only. This is understandable since the training and test set are more similar. Adding celebrities lowers the performances, but not significantly. We lose around 2% in each scenario, but still achieve 90% accuracy, which is a good result. We believe that adding more variance in the training set, as in this case, could help in a real-world situation with a lot of different people and backgrounds. Finally, using only celebrities to fine-tune the network leads to the worst accuracy, but still, the average is above 80%, which is remarkable since training and test set are very different. Comparing our results with the one state-of-the-art models (Table 3), we definitely reach much higher performance, even if the settings are partially different (i.e., we could not test the same people enrollment images since they served for our training). Against Azure v3, specifically built to detect blur faces, we score roughly 35% and 20% more on celebrities and people, respectively. Nonetheless, we remind the reader that our main purpose was not to surpass state-of-the-art algorithm, but develop an appropriate system that could detect blurred faces with high accuracy.



(a) *Scenario 1: Same person and fixed background*

(b) *Scenario 2: Different people and fixed background*

(c) *Scenario 3: Same person and variable background*

(d) *Scenario 4: Different people and variable background*

Fig. 9. Average logout accuracy (bars) and average grace period (dots) for different aggregation frames and application scenarios.

8.4. De-authentication performance

Face detection is the heart of BLUFADER. By detecting the user's face frame by frame, we are able to understand when they leave and de-authenticate them accordingly. Even with results above 90%, which are generally good in the computer vision area, we still need an improvement to provide users with a reliable de-authentication system. De-authenticate them every time the neural network fails the prediction is not desirable, and can negatively impact the users' experience. To improve BLUFADER, we can consider two crucial aspects: (i) the neural network commits sparse, not sequential mistakes, and (ii) the de-authentication has not to be instantaneous. In fact, the literature identifies a "grace period" in which the user might still be logged in even though they have already left. Obviously, this period must be short enough not to allow lunchtime attacks, and is based on the fact that users in that period can notice if someone is trying to steal their active session. A reasonable grace period is below six seconds [5].

Following these considerations, BLUFADER performs face detection and evaluates the results using a sliding window of aggregates frames. The de-authentication occurs once the face is not detected for N consecutive frames. N can be 1, meaning that at the first frame where the face is not detected, BLUFADER de-authenticates the user, or higher. In our experiments, we tested different values of N, to a maximum of 90, which means 3 s (the webcam recorded at 30 FPS). Fig. 9 shows the logout accuracy (i.e., the times BLUFADER correctly logs out a user) per different levels of N (aggregation frames) and the corresponding grace period needed to log out the user. The four graphs represent the four scenarios, and each bar in the plot represents a background accuracy, while the dots indicate the grace period. These graphs refer to the experiments using the people dataset only, which achieved better scores than using People and Celebrities or Celebrities only. We discuss these two cases later in this section. In general, the de-authentication accuracy trends reflect the underlying face detection system. For all the application scenarios, the accuracy increases as the aggregation does. Considering an aggregation frame equal to one, BLUFADER would wrongly de-authenticate users too frequently (i.e., over

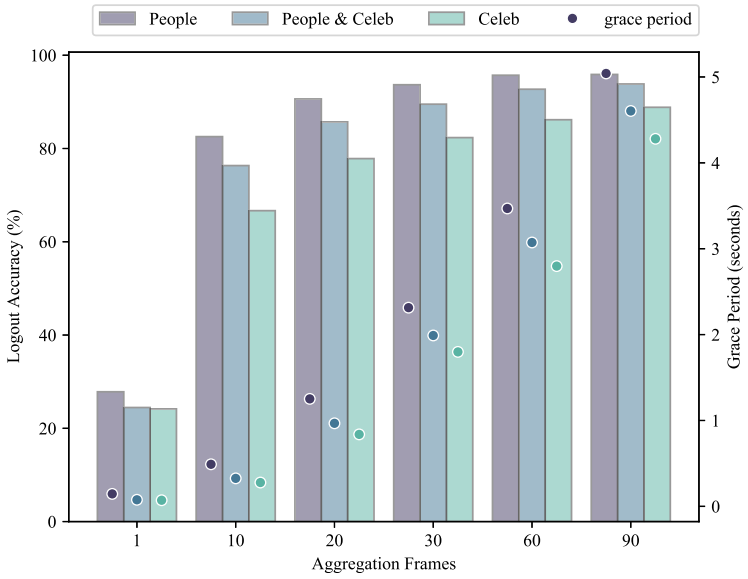


Fig. 10. Logout accuracy (bars) and grace period (dots) for different aggregation frames and training sets. The accuracy and grace periods are averaged on all the backgrounds and scenarios.

60% on average in all the scenarios and backgrounds), making our system unusable. On the other hand, considering a higher number of aggregation frames (i.e., 90 frames), the logout accuracy rate increases up to 100% for Scenario 1 (Fig. 9(a)) and scenario 2 (Fig. 9(b)) in b_1 and b_2, keeping the grace period under 5 s. Scenario 3 (Fig. 9(c)) shows the lowest performance of BLUFADER, with an accuracy below 80% even with 90 aggregation frames in b_4. However, the other backgrounds show very high scores with a grace period of under five seconds.

Considering all scenarios together, the difficulty of the backgrounds highly impacts the performances. The more complex the background, the less accuracy. Starting from 30 aggregation frames, b_1 reaches 100% of accuracy in all the scenarios, keeping the grace period below 3 s. b_2 shows similar performance, reaching 100% of accuracy in less than 4 s in all the scenarios when the aggregation frames is equal to 60. b_3 shows more than 95% accuracy with 90 aggregated frames in about five seconds, while b_4 struggles a bit, especially in the third scenario. These data reveal that BLUFADER can work incredibly well when the background is an empty wall or with simple decorations, like in a common work office, and struggles a bit with challenging backgrounds. However, when the background is fixed, BLUFADER always performs above 90%.

Fig. 10 compares the averaged BLUFADER performances in all the scenarios and background, with respect to the different training sets we used to fine-tune the network (i.e., People, People & Celebrities, Celebrities only). The plot clearly shows how adding more variance to the training set does not help the task. This is understandable since when using People only, the training and test set are more similar, which is preferable. In this case, BLUFADER achieves 96% accuracy in less than 4 s. On the other hand, when fine-tuning the network only using Celebrities, the training and test sets are very different. Still, BLUFADER achieves almost 90% accuracy in less than 5 s, which is remarkable.

9. Face recognition

The second core module of BLUFADER is face recognition. Given the high level of blur our filter provides, it is extremely difficult to create a model to map users' facial traits to their identity. As an alternative, we create a model that can tell whether two images represent the same person based on their (dis)similarity. In Section 9.1, we give the motivation of why a recognition module is needed, and we present it In Section 9.2. The face recognition performances and BLUFADER de-authentication performances are presented in Sections 9.3 and 9.4, respectively.

9.1. Security concerns of face detection

In Section 8, we showed that BLUFADER could accurately detect a blurred face, and de-authenticate the user when a face is not visible anymore. However, BLUFADER cannot recognize whose face it is. Such a limitation opens paths to security threats. For instance, if two faces appear simultaneously, how should the system behave? Currently, if a second person appears in the webcam's view and the logged user steps away simultaneously, BLUFADER would not terminate the session. The same applies if the attacker takes over the user's workstation within the grace period, which should be unlike to happen, but achievable in an opportunistic scenario (e.g., an accomplice gets the victim's attention and quickly

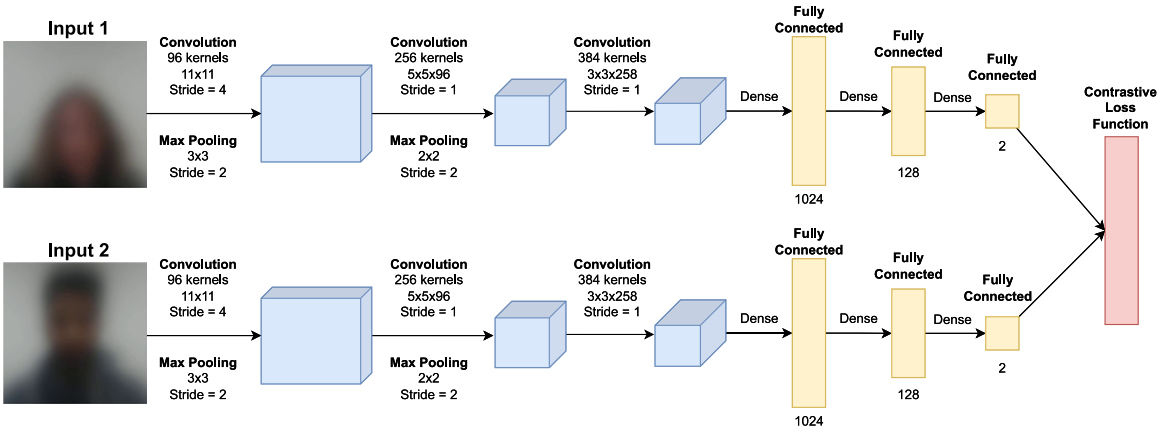


Fig. 11. Our Siamese Neural Network implemented for the face recognition stage. Blocks in blue represents the Convolutional Neural Networks responsible to extract visual features. In yellow, fully connected layers, while in red, the contrastive loss function used to train the network.

takes him away). Furthermore, given the high blur level, objects with a face-like shape could be recognized as faces by BLUFADER. Thus, if an attacker can put such an object in the user's background, a face would be recognized when the user steps away, keeping the session alive. All the problems we mentioned could be "easily" solved by adding a face recognition module to BLUFADER. Indeed, by "knowing" the face of the authenticated users, BLUFADER can correctly de-authenticate them every time an "unknown" face remains the only visible. In the following sections, we show how we developed such a recognition module.

9.2. Proposed face recognition model

Detecting faces accurately is the first step towards face recognition [41]. In Section 8.1, we demonstrated that state-of-the-art algorithms could not detect faces blurred with our filter, thus, they cannot recognize them either. Consequently, we developed our own face recognition algorithm that could cope with our filter's high blur level.

9.2.1. Overview

At the login phase (e.g., right after inserting a password), BLUFADER saves some "ground-truth" frames (i.e., the face) of the logging user. When BLUFADER detects a face (or more), it can be compared with the ground-truth frames to understand whether the user is the same, and, accordingly, whether to keep the user logged or not. To ensure the *permanence* of the method (i.e., overcome small perturbations happening over time), such ground-truth frames can be updated periodically. In our settings, to limit the computational costs of BLUFADER, we save 10 ground-truth frames, perform face recognition every two seconds or when multiple faces are detected, and update the ground-truth frames every minute, or when the results of (a successful) face detection reveal a significant change in the frames. The face recognition step leverages a Siamese Neural Network that, given two images, returns their dissimilarity scores. When the dissimilarity is low, the user is recognized and kept authenticated. When the dissimilarity overcomes a pre-determined threshold, BLUFADER de-authenticates the user.

9.2.2. Proposed model

The high level of blur provided by our *Antirefl* filter hinders the possibility of extracting many low-level filters to perform face recognition. Moreover, we cannot rely on thousand of images per person to recognize them since, in some scenarios, the workstation is shared. For such reasons, we designed a Siamese Neural Network (SNN), a paradigm that has been gaining traction in recent years [64]. Their success is largely due to their ability to scale with larger numbers of classes and their few sample requirements. Instead of learning the low-level features of all classes (which requires thousands of samples), they learn how to extract features fundamental for the task (e.g., classification, object recognition), and then provide a measure of dissimilarity. When two images are similar, they are labeled as belonging to the same class, and different classes otherwise. Such functioning is precisely what we need in our face recognition scenario. Instead of learning the low-level features of each user's face, we want a system able to tell when two faces belong to the same person. Having collected the ground-truth frames at the beginning of the session, we just need to check that the detected user is identical to the ground-truth user.

Fig. 11 represents the SNN we implemented to conduct face recognition. An SNN is made of two identical neural networks that work in tandem, sharing their weights. These networks are responsible for extracting the low-level representations of the inputs (i.e., their embeddings). Once these representations are computed, we can calculate their distance (e.g., using the Euclidean distance) to determine whether they are similar (i.e., belonging to the same class), or

not. Since we are working with images, we adopted a Convolutional Neural Network (CNN) to extract visual features. In particular, we applied the following:

1. Convolution2D: output channels = 96, kernel size = 11 × 11, stride = 4, activation = ReLu;
2. Max Pooling2D: kernel size = 3 × 3, stride = 2;
3. Convolution2D: output channels = 256, kernel size = 5 × 5, stride = 1, activation = ReLu;
4. Max Pooling2D: kernel size = 2 × 2, stride = 2;
5. Convolution2D: output channels = 384, kernel size = 3 × 3, stride = 1, activation = ReLu;

followed by three fully connected (FC) layers of 1024, 128, and 2 neurons, respectively. To train the network, we used the Contrastive Loss Function [65], which is defined as:

$$\mathcal{L}(A, B, Y) = (Y) * \|f(A) - f(B)\|^2 + (1 - Y) * max(0, m^2 - \|f(A) - f(B)\|^2), \quad (1)$$

where A and B are the two images pair, Y is the label of the pair (1 same class, 0 otherwise), $f(A)$ and $f(B)$ are the outputs of the network (i.e, after CNN and FC layers), and m is the margin. m is used to limit the distance between samples of different classes, and we set it to a standard value of 2. As a result of the Contrastive Loss Function, samples belonging to the same class will have a small distance (i.e., low dissimilarity), while samples belonging to different classes will have a considerable distance (i.e., high dissimilarity). Thus, when giving in input two frames, if the dissimilarity is low we conclude the person is the same in the two images, or different otherwise. We trained our SNN for our four scenarios using the settings explained in Section 7.2, following the same procedure of the face detection stage. We also enriched the enrollment snapshots with 60 frames (i.e., 2 s) coming from the enrollment videos. Once we created the scenario-dependent training set for each person in each background, we isolated 20% of the images for validation purposes. At each epoch, we feed the network N random pairs, half positive (i.e., same person, $Y = 1$) and half negative (i.e., different person, $Y = 0$), where N is the length of the training set. To prevent overfitting, the validation set was used to early-stop the network after five epochs of no loss decrement. We trained for a maximum of 100 epochs, with a batch size of 64, Adam optimizer, and a learning rate of 0.0005. To test our network, we performed for each person in each background two evaluations:

- **Logged User Recognition.** We calculated the dissimilarity between the 10 ground-truth frames of the logged users and 10 frames every two seconds of that user performing our four tasks (read, write, talk with colleague, look at smartphone) on the same background.
- **Different Users Recognition.** We calculated the dissimilarity between the 10 ground-truth frames of the logged users and 8 randomly selected frames (two frames per task) of all the people, but the logged one on the same background.

9.3. Face recognition performance

Dissimilarity distributions result significantly different among all four scenarios as depicted in Fig. 12. In particular, when the person is the same as the logged user, dissimilarity distribution assumes the low scale values (close to 0), while when a different person appears, the dissimilarity increase, reaching an average near 2.5. We can derive that distinguishing between the same and different users can be done with relatively high accuracy. Figs. 13 and 14 show examples of dissimilarities between frames of the same person and different people. Dissimilarity threshold selection to determine when the system recognizes the user is critical both for security and usability of BLUFADER. In particular, a low dissimilarity threshold leads to high false negative rates (i.e., the user is unrecognized too often), while increasing security. On the other hand, a high dissimilarity threshold causes an increase of false positives (i.e., different persons are recognized as the logged user), while improving usability. In Fig. 15, we show the impact of dissimilarity on F1-score, Recall, and Precision. Selecting a threshold of 0.5 (i.e., a compromise between security and usability), we report in Table 5 the scores of face recognition divided by scenarios, backgrounds, and tasks (T1 = read email, T2 = write sentence, T3 = look phone, T4 = talk with colleague, T5 = leave workstation). Generally, higher performance is obtained in the simpler background (i.e., b1). Surprisingly, there are no significant differences in accuracy between different backgrounds. This means that SNN succeeds in generalizing well, focusing mainly on the blurred faces rather than on changes in the background. In contrast, our approach seems particularly effective in scenario 1 where overall accuracy reaches 95% compared to the 88% average accuracy of the other three scenarios. T1 and T3 result in the tasks our model discerns most effectively, reaching values above 90% on average. As might be expected, due to the higher variability between ground-truth frames and randomly selected frames, our model shows lower performance in T4, especially in scenarios 2, 3, and 4.

9.4. De-authentication performance

Similarly to Section 8.4 even if face recognition results are encouraging, higher performance needs to be achieved to reach a better trade-off between security and usability. To this purpose, BLUFADER aggregates multiple frames in each face recognition session. In particular, we averaged the dissimilarity values for each ground-truth frames, filtering out possible inaccuracies of the SNN. As shown in Fig. 16, we notice a significant improvement in all the metrics. Indeed,

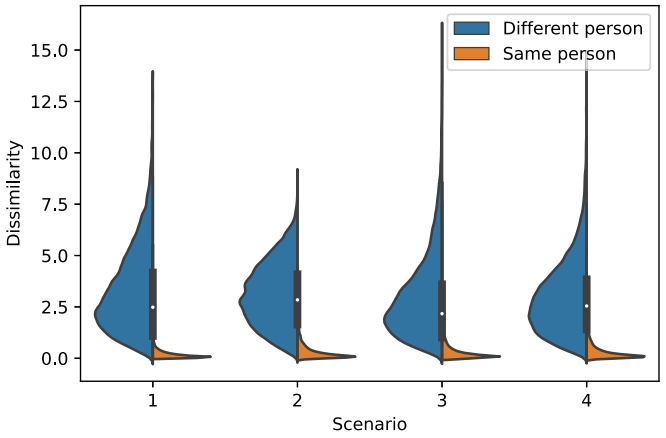
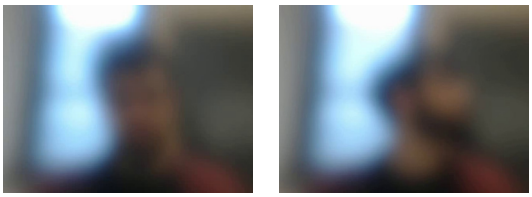
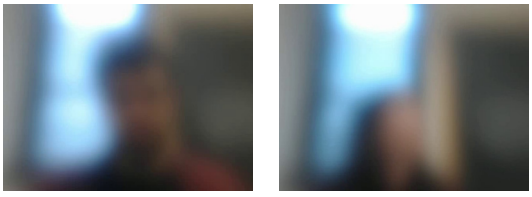


Fig. 12. Dissimilarity distributions in the four application scenarios.



(a) *Write* (b) *Talk with colleague*

Fig. 13. Frames from the same person on different tasks in background 4. The dissimilarity is 0.17.



(a) *Write* (b) *Talk with colleague*

Fig. 14. Frames from different subjects on different tasks in background 4. The dissimilarity is 2.63.

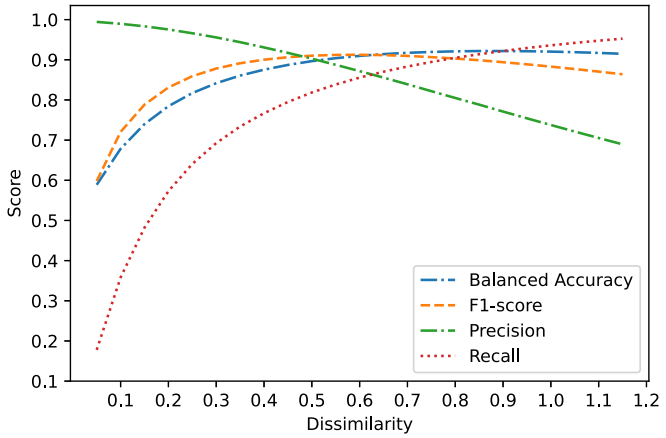


Fig. 15. Threshold evaluation through Balanced Accuracy, F1-score, Precision and Recall.

Table 5
Balanced accuracy (dissimilarity = 0.5) of face recognition divided by scenarios, backgrounds, training datasets, and tasks (T1 = read email, T2 = write sentence, T3 = look at phone, T4 = talk with colleague).

	Task	Scenario 1					Scenario 2					Scenario 3					Scenario 4				
		b1	b2	b3	b4	**Avg**	b1	b2	b3	b4	**Avg**	b1	b2	b3	b4	**Avg**	b1	b2	b3	b4	**Avg**
People	T1	0.99	0.99	0.97	0.98	**0.98**	0.93	0.95	0.95	0.96	**0.95**	0.91	0.96	0.96	0.95	**0.94**	0.93	0.95	0.95	0.97	**0.95**
	T2	0.93	0.88	0.93	0.89	**0.91**	0.79	0.78	0.81	0.75	**0.78**	0.78	0.79	0.81	0.78	**0.79**	0.78	0.78	0.79	0.75	**0.78**
	T3	0.97	0.96	0.95	0.98	**0.97**	0.91	0.91	0.91	0.92	**0.91**	0.90	0.92	0.90	0.94	**0.92**	0.92	0.91	0.90	0.94	**0.92**
	T4	0.96	0.94	0.96	0.96	**0.95**	0.86	0.87	0.88	0.84	**0.86**	0.86	0.88	0.88	0.86	**0.87**	0.86	0.88	0.87	0.85	**0.86**
	Overall	**0.96**	**0.94**	**0.95**	**0.95**	**0.95**	**0.87**	**0.88**	**0.89**	**0.87**	**0.88**	**0.86**	**0.89**	**0.89**	**0.88**	**0.88**	**0.87**	**0.88**	**0.88**	**0.88**	**0.88**

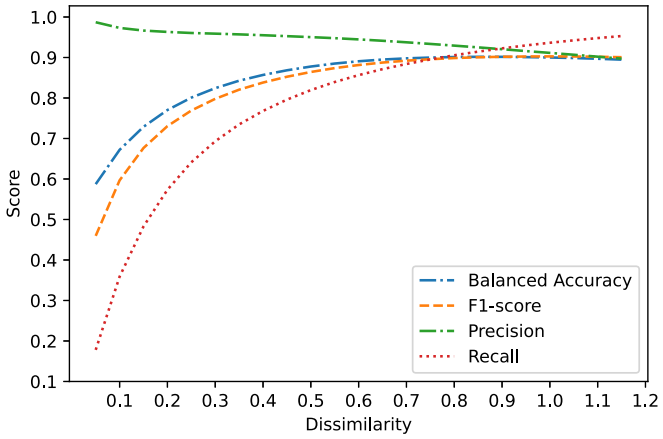


Fig. 16. Threshold evaluation through Balanced Accuracy, F1-score, Precision and Recall with aggregated results.

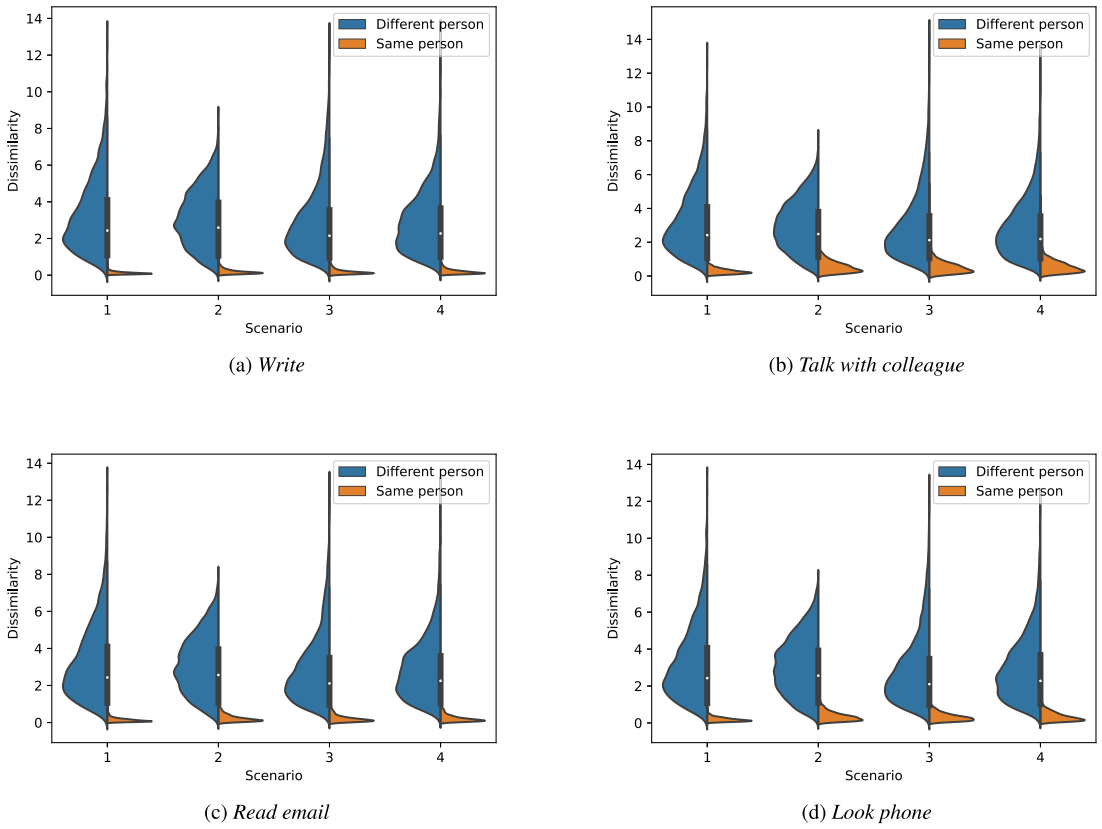
Table 6
Balanced accuracy (aggregated dissimilarity = 0.85) of face recognition divided by scenarios, backgrounds, training datasets, and tasks (T1 = read email, T2 = write sentence, T3 = look at phone, T4 = talk with colleague).

	Task	Scenario 1					Scenario 2					Scenario 3					Scenario 4				
		b1	b2	b3	b4	**Avg**	b1	b2	b3	b4	**Avg**	b1	b2	b3	b4	**Avg**	b1	b2	b3	b4	**Avg**
People	T1	0.99	0.99	0.96	0.98	**0.98**	0.97	0.96	0.97	0.97	**0.97**	0.95	0.96	0.95	0.97	**0.96**	0.98	0.98	0.95	0.97	**0.97**
	T2	0.98	0.98	0.96	0.97	**0.97**	0.87	0.88	0.91	0.85	**0.88**	0.86	0.91	0.89	0.87	**0.88**	0.90	0.87	0.86	0.83	**0.87**
	T3	0.99	0.98	0.95	0.98	**0.98**	0.96	0.95	0.95	0.97	**0.96**	0.95	0.96	0.91	0.97	**0.95**	0.97	0.96	0.94	0.98	**0.96**
	T4	0.98	0.98	0.95	0.98	**0.97**	0.92	0.93	0.96	0.92	**0.93**	0.94	0.96	0.93	0.94	**0.94**	0.95	0.94	0.94	0.94	**0.94**
	Overall	**0.99**	**0.98**	**0.96**	**0.98**	**0.98**	**0.93**	**0.93**	**0.95**	**0.93**	**0.93**	**0.92**	**0.95**	**0.92**	**0.94**	**0.93**	**0.95**	**0.94**	**0.92**	**0.93**	**0.93**

with a threshold of 0.85 precision and recall reach a score of 92%, improving the performance of 8% compared to the single frame approach. The improvement can also be noticed in Fig. 17. The new distributions are more skewed toward zero when the person is the same, and toward higher values when the person is different. The system performs better in scenarios 1 and 3, suggesting that including enrollment frames at training time helps in the face recognition stage. Looking at the tasks, "talk with colleague" is the one that presents a higher variance for dissimilarity between frames of the same users. This affects all the scenarios with slightly less effect on scenario 1. This means that by keeping the background constant and leveraging user enrollment frames, SNN results are more robust. The aggregated dissimilarity approach that we adopted increased the overall balanced accuracy by more than 5% compared to the single-frame approach. In particular, as reported in Table 6 all the scenarios show an overall accuracy higher than 93% up to 98% for scenario 1, demonstrating that this approach generalizes well among different backgrounds and scenarios.

10. Usability user study

We conducted an additional investigation regarding how users perceive the proposed de-authentication system. We recruited 27 participants and asked them to use a computer instrumented with BLUFADER for one hour. The population was distributed in 85,2% of males and 14,8% of females, with an age that spans from 23 to 40 and a standard deviation of 5,2. During the experiment, users could perform any action they wanted (e.g., reading news, writing emails, playing). Furthermore, we entertained the participants with some questions (simulating they would talk with a colleague), and asked them to take at least one break, leaving the workstation, to test the de-authentication mechanism. At the end of the experiment, we asked participants to fill out the widely adopted System Usability Scale (SUS) [66], a useful tool for evaluating perceived usability. Participants filled out the questionnaire twice: one for their experience with BLUFADER, and



(a) *Write* (b) *Talk with colleague*

(c) *Read email* (d) *Look phone*

Fig. 17. Dissimilarity distributions per task in the four application scenarios after aggregation.

one for their past experiences with an inactivity de-authentication system, which all participants were familiar with. We chose the inactivity method as a comparison against BLUFADER because it represents the easiest way to de-authenticate a user since it does not require any additional equipment or action from the user. Per each statement, the user had to rate the level of agreement expressing a value from 1 (*Strongly Disagree*) to 5 (*Strongly Agree*). Table 7 shows the questions of the SUS questionnaire,[10] while the results (average and standard deviation for each question) are plotted in Fig. 18.

An initial analysis from Fig. 18 shows that the two methodologies are not so far apart in terms of usability. This is already a positive result for BLUFADER, since de-authentication for inactivity requires no actions from the users, and is widely adopted by most modern operating systems and devices. Following the *standard SUS calculation methodology* for both the de-authentication systems, we calculated the SUS scores to support the previous intuition. The value is expressed on a 0 (*completely unusable*) to 100 (*perfectly usable*) scale. With a score of 71.25, BLUFADER surpasses *Inactivity*, which scored 68.2 instead. The *Adjective Rating Scale* [67] offers an additional and yet more comprehensive outcome of the analysis: on a scale of seven adjectives,[11] BLUFADER falls within the *"Good"* label, while *Inactivity* obtains the lower level adjective (i.e., *"Ok"*). On an *acceptability scale*, presented in [67], both the methodologies' results are acceptable, even though the *Inactivity* is only *"marginally acceptable"* since it falls within the score range of 50–70. Further considerations can be drawn based on the results obtained from the individual questions (Fig. 18). Despite the *enrollment procedure* that BLUFADER might need, users rated it as less complex to use (question *A2*, *A4* and *A8* in Table 7). As previously discussed, these results might arise from the intrinsic difficulties in setting appropriate inactivity timers [2]. The functions offered by BLUFADER are well integrated (*A5*), and above all, it appears less inconsistent in its overall design than inactivity (*A6*). The user study also suggests that BLUFADER is easier to learn (*A7*, *A10*). Not surprisingly, BLUFADER surpassed the inactivity approach in terms of confidence (*A9*). We assume such achievement comes from the BLUFADER's reliability, privacy-preserving characteristics, and low errors. Furthermore, we evaluated the statistical significance of the singular queries of the SUS questionnaire through *unpaired t-test*. Results highlight that BLUFADER is statistically superior (p-*value* < 0.05) only regarding *A10* question, and comparable otherwise. To sum up, BLUFADER usability was rated similarly or mostly better than inactivity timeouts, which is the default mechanism in modern devices. Given its superior performance

10 Participants were also invited to report any additional system issues not addressed in the questionnaire through a non-mandatory open question.

11 https://measuringu.com/interpret-sus-score/.

Table 7
Questions contained in the SUS questionnaire [66].

Code	Question
A1	*I think that I would like to use this system frequently.*
A2	*I found the system unnecessarily complex.*
A3	*I thought the system was easy to use.*
A4	*I think that I would need the support of a technical person to be able to use this system.*
A5	*I found the various functions in this system were well integrated.*
A6	*I thought there was too much inconsistency in this system.*
A7	*I would imagine that most people would learn to use this system very quickly.*
A8	*I found the system very cumbersome to use.*
A9	*I felt very confident using the system.*
A10	*I needed to learn a lot of things before I could get going with this system.*

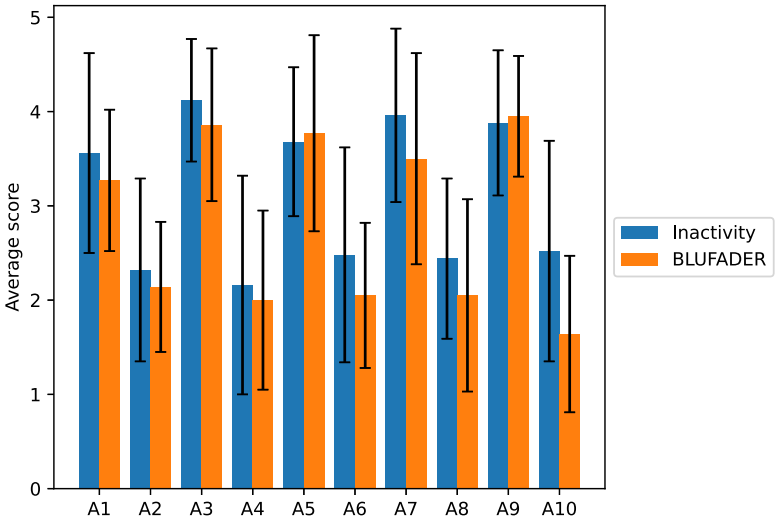


Fig. 18. Average scores of the user study with error bars (standard deviation).

and advantages with respect to other de-authentication systems, we believe it could become a practical de-authentication system for everyday use.

10.1. Performance analysis

BLUFADER should run continuously in real-time to function properly. Therefore, it is important to measure the computational costs of the implementation. When developing our model, we considered this factor by choosing a fast and accurate state-of-the-art object detection model (i.e., RetinaNet [60], as demonstrated by the authors) to fine-tune. When considering the computational costs required for the model, we remind the reader that the inference phase (i.e., making the prediction), is much faster and less resource-demanding than training the model. Indeed, while the loss optimization in the training phase requires having in memory all the model weights and activations, as well as calculating derivatives and updating all the parameters (i.e., back-propagation), the inference phase requires only a forward pass and keep in the memory only two layers at a time, since the rest, once calculated, can be discarded. This results in minimal hardware requirements. For instance, a real-time object detector using ResNet50 (the backbone of RetinaNet) can be easily deployed on a Raspberry Pi 4 with no GPU [68].

In our experiments, we tested BLUFADER on a workstation equipped with AMD Ryzen 5 3600x 6-Core 12-Threads CPU, 32 GB RAM, and RTX 3090 GPU. To simulate a less powerful scenario, we limited the execution to a single core and a maximum of 2 GB GPU memory.[12] BLUFADER model occupies 128 MB of disk memory, and the process requires roughly 1.28 GB of RAM. With these specifications, we could analyze a maximum of 48 FPS, well above the 30 FPS provided by the webcam. During the testing, the participants did not report any PC slowdown, as also demonstrated by the low scores of answers *A6* and *A8*. Even though not everyone has a GPU to ensure high performance, we argue it might not be necessary at all. First, an average CPU can still perform real-time object detection [68] if the model is properly optimized

[12] Please note that every modern computer has at least two cores since the production of single-core desktop processors ended in 2013 with the Celeron G470 [69].

(our is currently not, which is a limitation we plan to overcome in the future). Second, given BLUFADER privacy-preserving capability (see Section 5), the frames can be sent to a server for analysis, reducing the overall costs down to a web request. Indeed, as specified in our threat model, we aim to remove the trust in the manufacturer (which could have access to the webcam) by physically blurring the video stream, making the person identification impossible. Last, researchers further reduced RetinaNet computational requirements by developing lighter (but equally performing) models [70,71]. Although precise testing should be conducted in the future, we believe such models could fit our scenarios too, further reducing BLUFADER requirements.

10.2. Limitations

Though BLUFADER achieves good performance, it has some limitations. First, our participants set includes a few ethnicities, and subjects were tested in just four backgrounds. We added more variance using the celebrities dataset, and the good results suggest BLUFADER would work even with different people. Still, more evaluations need to be conducted. Nonetheless, the four scenarios give us a good idea of how BLUFADER would work in the real world. Second, participants performed their tasks for ten seconds each. Clearly, longer use of BLUFADER needs to be evaluated. Third, each user involved in the usability evaluation tested the system for one hour: this time window is relevant for an initial assessment, but it would be beneficial to extend such a period. This way, it will be possible to reproduce a better approximation of the machine usage. We evaluated BLUFADER's usability through direct testing, but Inactivity one was based on user recall. Although this evaluation difference may affect the systems' comparison, we argue that most users adopt inactivity de-authentication mechanisms embedded in popular operating systems and devices. In fact, participants were required to be familiar with or regularly use inactivity timeouts. Users did not raise any concern in the usability study; however, BLUFADER performance can be significantly improved. Future developments of the proposed system will allow reducing hardware consumption requirements. Consequently, the model will be potentially usable on a broader set of older and less powerful machines without compromising the user's session.

11. Conclusions

In this work, we presented BLUFADER, a de-authentication system based on blurred face detection and recognition deep learning algorithms. We conducted extensive experiments to select the physical blurring material for BLUFADER, to remove facial traits, ensuring privacy, while allowing face detection by deep learning algorithms. Users' privacy was evaluated through an online survey, demonstrating that a simple anti-reflex tape applied to the webcam is sufficient to make a face unrecognizable. Further, we demonstrated that our blurring material is resistant to deblurring state-of-the-art algorithms. By continually detecting the users' blurred faces, BLUFADER automatically de-authenticates them with very high accuracy, i.e., up to 100% in under 3 s on simple backgrounds, or 96% within 4 s considering also difficult ones. We tested BLUFADER in four scenarios that represent most of the real-world systems, ranging from laptops to ATMs, with 30 people conducting five different tasks. Our face detection neural network outperforms both commercial and literature state-of-the-art algorithms, demonstrating that fine-tuning can help in the detection of highly blurred objects and faces. Moreover, we implemented a face recognition module to distinguish users from their blurred images, protecting them from several adversarial attacks and enhancing overall security. We let 27 users test the developed system, collecting their impressions. The latter results demonstrate the real applicability of the model in a real scenario. The high results demonstrate that BLUFADER is fast, secure, and effectively preserves users' privacy.

11.1. Ethical concerns

Our institutions do not require any formal IRB approval to carry out the experiments described herein. Nonetheless, our surveys, experiments, and corresponding evaluation are all performed by adhering to the guidelines of the Menlo report [72]. All voluntary participants were informed that their data and responses would be used for research purposes. Participants are also aware of the email address to contact if they wish to have their entries removed from our dataset. Since our user base is located in Europe, we also strictly complied with the GDPR. All voluntary participants were informed of the actual use of their data, and their informed consent was obtained before the recording process. All the data have been anonymized and used by the authors of this paper for research purposes only.

Declaration of competing interest

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

Data availability

We have publicly released our datasets.

Appendix A. Celebrities privacy survey structure

This appendix provides a representative part of the submitted online questionnaire to evaluate the effectiveness of the blurring physical filters. Since the questionnaire is repetitive, we report only one of the 10 celebrities involved in the survey. In Section 5.2, we discuss the motivations and results of the questionnaire.

Do you know them?

This is a simple game to test privacy on images. We show you a photo of a person, and you guess the name. **Please, once you have inserted the name, DO NOT go back and change it, thanks!** It should take no more than 3 minutes to complete. If you have no idea who the person is, answer "no idea".

Please Insert your age:

_____

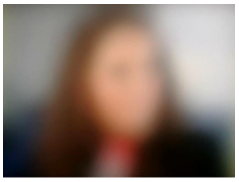
Please Insert your gender:

- Male
- Female
- Prefer not to say

Please Insert your email:

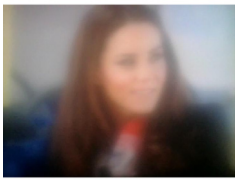
_____

Write the name:



_____

Write the name:



_____

Write the name:

_____

Appendix B. Deblurred celebrities privacy survey structure

Based on Appendix A, the survey for the quality evaluation of the deblurring models asks to the respondents to recognize 10 celebrities (different from the ones selected in Appendix A) with different filters applied. A deep discussion about this survey can be found in Section 6.2.

Do you know them?

This is a simple game to test privacy on images. We show you a photo of a person, and you guess the name. **Please, once you have inserted the name, DO NOT go back and change it, thanks!** It should take no more than 3 minutes to complete. If you have no idea who the person is, just answer "no idea".

Please Insert your age:

_____

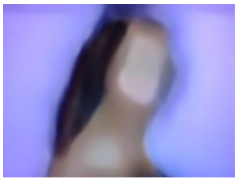
Please Insert your gender:

- Male
- Female
- Prefer not to say

Please Insert your email:

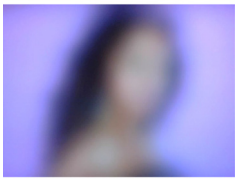
_____

Write the name:



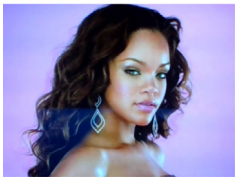
_____

Write the name:



_____

Write the name:



_____

References

[1] S. Eberz, K. Rasmussen, V. Lenders, I. Martinovic, Preventing lunchtime attacks: Fighting insider threats with eye movement biometrics, in: 22th NDSS 2015, Internet Society, 2015.

[2] S. Sinclair, S.W. Smith, Preventative directions for insider threat mitigation via access control, in: Insider Attack and Cyber Security, Springer, 2008, pp. 165–194.

[3] U. Mahbub, V.M. Patel, D. Chandra, B. Barbello, R. Chellappa, Partial face detection for continuous authentication, in: 2016 IEEE ICIP, IEEE, 2016, pp. 2991–2995.

[4] S. Mare, A.M. Markham, C. Cornelius, R. Peterson, D. Kotz, Zebra: Zero-effort bilateral recurring authentication, in: 2014 IEEE Symposium on Security and Privacy, IEEE, 2014, pp. 705–720.

[5] M. Conti, G. Lovisotto, I. Martinovic, G. Tsudik, Fadewich: fast deauthentication over the wireless channel, in: 2017 IEEE 37th ICDCS, IEEE, 2017, pp. 2294–2301.
[6] I. Masi, Y. Wu, T. Hassner, P. Natarajan, Deep face recognition: A survey, in: 2018 31st SIBGRAPI, IEEE, 2018.
[7] S.P. Banerjee, D.L. Woodard, Biometric authentication and identification using keystroke dynamics: A survey, J. Pattern Recognit. Res. 7 (1) (2012) 116–139.
[8] S. Hanisch, P. Arias-Cabarcos, J. Parra-Arnau, T. Strufe, Privacy-protecting techniques for behavioral data: A survey, 2021, arXiv preprint arXiv:2109.04120.
[9] J. Tolsdorf, D. Reinhardt, L.L. Iacono, Employees' privacy perceptions: exploring the dimensionality and antecedents of personal data sensitivity and willingness to disclose, Proc. Priv. Enhanc. Technol. 2022 (2) (2022) 68–94.
[10] M. Brocker, S. Checkoway, ISeeYou: Disabling the MacBook webcam indicator LED, in: 23rd USENIX, 2014, pp. 337–352.
[11] D. Machuletz, H. Sendt, S. Laube, R. Böhme, Users protect their privacy if they can: Determinants of webcam covering behavior, in: Proceedings of EuroSEC'16, 2016.
[12] J. Hattem, FBI director: Cover up your webcam, 2016, https://thehill.com/policy/national-security/295933-fbi-director-cover-up-your-webcam, Accessed: September, 2021.
[13] M. Cardaioli, M. Conti, P.P. Tricomi, G. Tsudik, Privacy-friendly De-authentication with BLUFADE: Blurred face detection, in: 2022 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, 2022, pp. 197–206.
[14] H.D. Company, HP presence aware, 2020, https://tinyurl.com/HPunattended, Accessed: October, 2021.
[15] D. Marques, I. Muslukhov, T. Guerreiro, L. Carriço, K. Beznosov, Snooping on mobile phones: Prevalence and trends, in: Twelfth SOUPS 2016), 2016.
[16] P. Samangouei, V.M. Patel, R. Chellappa, Facial attributes for active authentication on mobile devices, Image Vis. Comput. 58 (2017) 181–192.
[17] G. Peng, G. Zhou, D.T. Nguyen, X. Qi, Q. Yang, S. Wang, Continuous authentication with touch behavioral biometrics and voice on wearable glasses, IEEE Trans. Hum.-Mach. Syst. 47 (3) (2016) 404–416.
[18] R. Damaševičius, R. Maskeliūnas, A. Venčkauskas, M. Woźniak, Smartphone user identity verification using gait characteristics, Symmetry 8 (10) (2016) 100.
[19] C. Shen, T. Yu, S. Yuan, Y. Li, X. Guan, Performance analysis of motion-sensor behavior for user authentication on smartphones, Sensors 16 (3) (2016) 345.
[20] M. Conti, P.P. Tricomi, PvP: Profiling versus player! Exploiting gaming data for player recognition, in: International Conference on Information Security, Springer, 2020, pp. 393–408.
[21] S. Ayeswarya, J. Norman, A survey on different continuous authentication systems, Int. J. Biometrics 11 (1) (2019) 67–99.
[22] L. Hernández-Álvarez, J.M. de Fuentes, L. González-Manzano, L. Hernández Encinas, Privacy-preserving sensor-based continuous authentication and user profiling: a review, Sensors 21 (1) (2021) 92.
[23] C.M. TEY, P. GUPTA, D. GAO, I can be you: Questioning the use of keystroke dynamics as biometrics.(2013), in: 20th NDSS 2013, 2013, pp. 1–16.
[24] M.D. Corner, B.D. Noble, Zero-interaction authentication, in: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, ACM, 2002, pp. 1–11.
[25] O. Huhta, P. Shrestha, S. Udar, M. Juuti, N. Saxena, N. Asokan, Pitfalls in designing zero-effort deauthentication: Opportunistic human observation attacks, in: NDSS, 2016.
[26] K.B. Rasmussen, M. Roeschlin, I. Martinovic, G. Tsudik, Authentication using pulse- response biometrics, in: NDSS, 2014.
[27] T. Kaczmarek, E. Ozturk, G. Tsudik, Assentication: User de-authentication and lunchtime attack mitigation with seated posture biometric, in: International Conference on Applied Cryptography and Network Security, Springer, 2018, pp. 616–633.
[28] M. Conti, P.P. Tricomi, G. Tsudik, DE-auth of the blue! Transparent de-authentication using bluetooth low energy beacon, in: ESORICS, Springer, 2020, pp. 277–294.
[29] D. Crouse, H. Han, D. Chandra, B. Barbello, A.K. Jain, Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data, in: 2015 ICB, 2015, pp. 135–142, http://dx.doi.org/10.1109/ICB.2015.7139043.
[30] P. Perera, V.M. Patel, Face-based multiple user active authentication on mobile devices, IEEE Trans. Inf. Forensics Secur. 14 (5) (2018) 1240–1250.
[31] T. Kanade, Picture processing system by computer complex and recognition of human faces, 1974.
[32] R. Brunelli, T. Poggio, Face recognition: Features versus templates, IEEE Trans. Pattern Anal. Mach. Intell. 15 (10) (1993) 1042–1052.
[33] P.J. Phillips, A.J. O'toole, Comparison of human and computer performance across face recognition experiments, Image Vis. Comput. 32 (1) (2014) 74–85.
[34] I. Gruber, M. Hlaváč, M. Železnỳ, A. Karpov, Facing face recognition with ResNet: Round one, in: International Conference on Interactive Collaborative Robotics, Springer, 2017, pp. 67–74.
[35] B.-N. Kang, Y. Kim, D. Kim, Pairwise relational networks for face recognition, in: Proceedings of ECCV, 2018, pp. 628–645.
[36] X. Lu, Y. Yang, W. Zhang, Q. Wang, Y. Wang, Face verification with multi-task and multi-scale feature fusion, Entropy 19 (5) (2017) 228.
[37] U. Zafar, M. Ghafoor, T. Zia, G. Ahmed, A. Latif, K.R. Malik, A.M. Sharif, Face recognition with Bayesian convolutional networks for robust surveillance systems, EURASIP J. Image Video Process. 2019 (1) (2019) 1–10.
[38] L. Song, D. Gong, Z. Li, C. Liu, W. Liu, Occlusion robust face recognition based on mask learning with pairwise differential siamese network, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 773–782.
[39] H. Wu, Z. Xu, J. Zhang, W. Yan, X. Ma, Face recognition based on convolution siamese networks, in: 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), IEEE, 2017, pp. 1–5.
[40] R. Jafri, H.R. Arabnia, A survey of face recognition techniques, J. Inf. Process. Syst. 5 (2) (2009) 41–68.
[41] G. Guo, N. Zhang, A survey on deep learning based face recognition, Comput. Vis. Image Underst. 189 (2019) 102805.
[42] M.-H. Yang, D.J. Kriegman, N. Ahuja, Detecting faces in images: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 24 (1) (2002) 34–58.
[43] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: A literature survey, ACM CSUR 35 (4) (2003).
[44] S. Yang, P. Luo, C.C. Loy, X. Tang, Faceness-net: Face detection through deep facial part responses, IEEE Trans. Pattern Anal. Mach. Intell. 40 (8) (2017) 1845–1859.
[45] G. Borghi, M. Venturelli, R. Vezzani, R. Cucchiara, Poseidon: Face-from-depth for driver pose estimation, in: Proceedings of the IEEE CVPR, 2017, pp. 4661–4670.
[46] Y. Zhu, H. Cai, S. Zhang, C. Wang, Y. Xiong, Tinaface: Strong but simple baseline for face detection, 2020, arXiv:2011.13183.
[47] S. Yang, P. Luo, C.-C. Loy, X. Tang, Wider face: A face detection benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5525–5533.
[48] S. Zafeiriou, C. Zhang, Z. Zhang, A survey on face detection in the wild: past, present and future, Comput. Vis. Image Underst. 138 (2015) 1–24.
[49] Y. Zhou, D. Liu, T. Huang, Survey of face detection on low-quality images, in: 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018), 2018, pp. 769–773, http://dx.doi.org/10.1109/FG.2018.00121.
[50] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, Pattern Recognit. 84 (2018) 317–331.
[51] A. Mittal, R. Soundararajan, A.C. Bovik, Making a "completely blind" image quality analyzer, IEEE Signal Process. Lett. 20 (3) (2012) 209–212.

[52] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of ICCV, 2015.
[53] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, A. Torralba, SUN database: Large-scale scene recognition from abbey to zoo, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 3485–3492, http://dx.doi.org/10.1109/CVPR.2010.5539970.
[54] K. Zhang, W. Ren, W. Luo, W. Lai, B. Stenger, M. Yang, H. Li, Deep image deblurring: A survey, Int. J. Comput. Vis. 130 (9) (2022) 2103–2130, http://dx.doi.org/10.1007/s11263-022-01633-5.
[55] S. Cho, S. Ji, J. Hong, S. Jung, S. Ko, Rethinking coarse-to-fine approach in single image deblurring, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, IEEE, 2021, pp. 4621–4630, http://dx.doi.org/10.1109/ICCV48922.2021.00460.
[56] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, Y. Li, MAXIM: Multi-axis MLP for image processing, 2022, CoRR, abs/2201.02973.
[57] X. Mao, Y. Liu, W. Shen, Q. Li, Y. Wang, Deep residual Fourier transformation for single image deblurring, 2021, CoRR, abs/2111.11745.
[58] S. Nah, T.H. Kim, K.M. Lee, Deep multi-scale convolutional neural network for dynamic scene deblurring, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, 2017, pp. 257–265, http://dx.doi.org/10.1109/CVPR.2017.35.
[59] I. Vasiljevic, A. Chakrabarti, G. Shakhnarovich, Examining the impact of blur on recognition by convolutional networks, 2016, arXiv preprint arXiv:1611.05760.
[60] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[61] TensorFlow, Eager few shot object detection colab, 2020, https://tinyurl.com/FineTuningTF, Accessed: January, 2021.
[62] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014.
[63] A. Boukerche, M. Sha, Design guidelines on deep learning–based pedestrian detection methods for supporting autonomous vehicles, ACM Comput. Surv. 54 (6) (2021) 1–36.
[64] D. Chicco, Siamese neural networks: An overview, Artif. Neural Netw. (2021) 73–94.
[65] F. Wang, H. Liu, Understanding the behaviour of contrastive loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2495–2504.
[66] J. Brooke, Usability evaluation in industry, chap. SUS: a "quick and dirty" usability scale, 1996.
[67] A. Bangor, P. Kortum, J. Miller, Determining what individual SUS scores mean: Adding an adjective rating scale, J. Usability Stud. 4 (3) (2009) 114–123.
[68] MathWorks, Identify objects within live video using ResNet-50 on raspberry pi hardware, 2022, https://www.mathworks.com/help/supportpkg/raspberrypiio/ref/identify-objects-within-video-using-resNet-50-on-raspberry-pi-hardware.html, Accessed: December, 2022.
[69] A. Computers, The last single core CPU, 2021, http://www.andyscomputer.net/2021/01/the-last-single-core-cpu.html, Accessed: December, 2022.
[70] Y. Li, A. Dua, F. Ren, Light-weight RetinaNet for object detection on edge devices, in: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), 2020, pp. 1–6, http://dx.doi.org/10.1109/WF-IoT48130.2020.9221150.
[71] M. Cheng, J. Bai, L. Li, Q. Chen, X. Zhou, H. Zhang, P. Zhang, Tiny-RetinaNet: a one-stage detector for real-time object detection, in: Eleventh International Conference on Graphics and Image Processing (ICGIP 2019), Vol. 11373, SPIE, 2020, pp. 195–202.
[72] M. Bailey, D. Dittrich, E. Kenneally, D. Maughan, The menlo report, IEEE Secur. Priv. 10 (2) (2012) 71–75.