Privacy-Friendly De-Authentication with BLUFADE

Blurred Face Detection

Cardaioli, Matteo; Conti, Mauro; Tricomi, Pier Paolo; Tsudik, Gene

DOI
10.1109/PerCom53586.2022.9762394

Publication date
2022

Document Version
Final published version

Published in
2022 IEEE International Conference on Pervasive Computing and Communications, PerCom 2022

2022 IEEE International Conference on Pervasive Computing and Communications (PerCom) | 978-1-6654-1643-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/PerCom53586.2022.9762394

Privacy-Friendly De-authentication with BLUFADE: Blurred Face Detection

Matteo Cardaioli*‡, Mauro Conti*§, Pier Paolo Tricomi*, Gene Tsudik†
*Department of Mathematics, University of Padua, Padua, Italy
{cardaiol, conti, tricomi}@math.unipd.it
†Department of Computer Science, University of California, Irvine, USA
gene.tsudik@uci.edu
‡GFT Italy, Milan, Italy
§Delft University of Technology, Delft, The Netherlands

***Abstract*—Ideally, secure user sessions should start and end with authentication and de-authentication phases, respectively. While the user must pass the former to start a secure session, the latter's importance is often ignored or underestimated. Dangling or unattended sessions expose users to well-known *Lunchtime Attacks*. To mitigate this threat, the research community focused on automated de-authentication systems. Unfortunately, no single approach offers security, privacy, and usability. For instance, although facial recognition-based methods might be a good fit for security and usability, they violate user privacy by constantly recording the user and the surrounding environment.**

In this work, we propose BLUFADE, a fast, secure, and transparent de-authentication system that takes advantage of blurred faces to preserve user privacy. We obfuscate a webcam with a physical blur layer and use deep learning algorithms to perform face detection continuously. To assess BLUFADE's practicality, we collected two datasets formed by 30 recruited subjects (users) and thousands of physically blurred celebrity photos. The former was used to train and evaluate the de-authentication system performances, the latter to assess the privacy and to increase variance in training data. We show that our approach outperforms state-of-the-art methods in detecting blurred faces, achieving up to 95% accuracy. Furthermore, we demonstrate that BLUFADE effectively de-authenticates users up to 100% accuracy in under 3 seconds, while satisfying security, privacy, and usability requirements.

***Index Terms*—De-authentication, Lunchtime Attacks, Privacy, Usability, Deep Learning, Blurred Face Detection**

I. Introduction

To begin using any modern computing device (e.g., desktop, workstation, laptop, tablet, or smartphone), the user must be authenticated. During the authentication process, the user is typically asked to demonstrate possession or knowledge of one or more of: (1) a secret, such as a password or PIN, (2) a biometric, such as a face or fingerprint, and (3) a device, such as a secure dongle or smartphone. Massive investments were made over the years to create and support secure means of user authentication.

At a later time, when the user ends (or abandons) its current session on a logged-in device, so-called *de-authentication* must ideally take place. However, in contrast with authentication, de-authentication received substantially less attention since lack thereof is not perceived as necessary as lack of (or insufficient) authentication. This is unfortunate since an unattended active secure session triggers the very real danger of *Lunchtime Attacks* [15]. 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.

This motivates the need for secure, privacy-preserving, and usable de-authentication techniques. However, prior results do not satisfy all these three requirements. For instance, the popular means of de-authentication via inactivity timeouts can be considered somewhat[1] privacy-preserving. However, if timeouts are too long, it offers poor security as the lunchtime attack time window grows. Whereas, if timeouts are too short, usability suffers since the user might need to re-authenticate needlessly [40]. Other methods continuously authenticate the user, and de-authentication occurs once the user's identity can no longer be verified. Common techniques rely on detecting physical presence of the user [8], [30], [31].

We believe that continuous face recognition is a promising means of de-authentication. It tracks and identifies previously authenticated user's face as long as it is visible from the webcam; once the user's face disappears from view (for a specific time interval), de-authentication occurs. This general approach offers several benefits. First, it is easy to implement and does not require extra equipment since most modern general-purpose computing devices are equipped with video cameras. Second, it is secure because current face detection algorithms are fast and highly accurate [33], making it resistant to *Lunchtime Attacks*. Third, it keeps the user authenticated and logged in, even if keyboard or mouse activity stops, as long as the user's face remains within line-of-sight of the webcam. This is in contrast with methods based on inactivity intervals, keystroke dynamics [2] or gaze–tracking [15], where users have to interact with the system continuously or frequently.

However, face recognition in de-authentication is hampered by significant **privacy concerns**. First, most users would not want to be video-recorded continuously. Even if the rules explicitly state that recordings are not stored anywhere, users might (rightfully) not trust such promises and refrain from

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

(or attempt to circumvent) using such a method. Second, an attacker who gains access to the webcam or recordings could exploit this information for malicious purposes. Blackmailing a user recorded during private moments is just one of many possible threats.

Nonetheless, most modern devices are equipped with user-facing cameras, and despite the manufacturers' assurances that cameras only operate in tandem with some user-visible indicator (e.g., an LED light in, or next to, the camera), many users find the constant presence of the camera unnerving. In fact, on some computers with integrated cameras, it is possible to surreptitiously turn on the camera and record **without** triggering the obligatory indicator [5].

Due to privacy and safety concerns, many cautious users have been applying physical barriers (e.g., placing tape) on their webcams [29]. This practice was publicly supported by the ex-FBI director James Comey [18], and some manufacturers now deliver laptops with built-in sliders to cover webcams.

Motivated by the above discussion, we propose `BLUFADE`, a de-authentication system based on continuous face detection that provides user privacy, security, and usability. We apply a physical blurring material on the webcam that obfuscates users' facial traits, making them unrecognizable. Then, after demonstrating that state-of-the-art face detection models perform very poorly on blurred images, we implemented a deep neural network for this specific task. We tested our system with 30 subjects in different scenarios and activities, reaching over 95% detection accuracy.

Contributions:

- A novel secure, usable, and privacy preserving de-authentication method based on blurred face detection;
- Its evaluation via extensive experiments, demonstrating that it outperforms state-of-the-art algorithms on blurred face detection tasks;
- Publicly released two datasets of physically blurred faces: the first one consists of $20k$ images of celebrities and backgrounds, blurred with two different materials, and the second contains $1,080$ enrollment images and 600 videos of 30 subjects interacting with a laptop (both blurred).

Organization: Section II overviews related work. Next, Section III describes the model, followed by Section IV and Section V which discuss the material evaluation and selection, respectively. Then, Section VI describes our experiments. Results are reported and discussed in Section VII, and Section VIII concludes the paper.

II. Related Work

Related work stems from several areas, including de-authentication as well as face recognition and detection.

A. 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 [7]). However, as mentioned earlier, failure to de-authenticate opens the door for lunchtime attacks, which are pretty common, as noted by Marques et al. [32]. 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 [40]. 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 recognition of: face [30], [38], voice [34], motion [13], [39], keystroke and/or mouse dynamics [3], and even video-game playing style [9]. For an extensive list of these techniques, we refer to [1], [19].

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 [42]. Carrying around a unique token that communicates with the workstation is another option [11]. However, its prominent drawback is the requirement to always carry and protect this token. A similar approach is explored in ZEBRA [31]: the user is continuously authenticated using a personal bracelet as long as wrist movements and the computer actions match. Unfortunately, [20] showed that Zebra is insecure. More complex and exotic systems, e.g., based on gaze–tracking [15] and pulse–response [37] 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 [8] instruments an office with position sensors to detect whether the users are sitting at their desks. *Assentication* [22] detects user presence through pressure sensors in the chair cushion. Whereas, [10]

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 [12], [35], [38]. In this paper, we focus on detection – rather than recognition – of faces, since most facial features would not be visible for privacy reasons. Since the user is already logged in, it is enough to trace the presence (detection) of their face.

B. 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 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 [6], [23]. With the rise of deep learning, especially Convolutional Neural Networks (CNN), computers reached (and surpassed) human performance in such tasks [36]. Deep-learning-based face recognition techniques can be divided into: (1) ones using single CNN [16], [24], (2) multi CNNs [28], and (3) variants of CNN [48]. For a comprehensive list of face recognition methods, refer to [17], [21].

Similar to face recognition, early face detection methods were based on developing discriminative hand-crafted features from faces and building robust learning algorithms [45], [50]. Nowadays, with the evolution of CNNs, detecting frontal faces is considered a solved task [33]. More efforts took place to detect faces under challenging conditions, such as partial faces [47] or faces captured by depth sensors [4]. Recently, TinaFace [52], 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 [46]. We refer to [49] for a complete treatment of this topic. Finally, [51] 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.

III. Model Overview

We now describe our system model and its real-world application scenarios.

A. 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, BLUFADE 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 BLUFADE's goal is to detect, and not to recognize, faces since the tape should blur the image enough to obscure facial traits.

Besides privacy, BLUFADE 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, and (ii) developing an algorithm to detect faces from blurred images. (i) is analyzed in Section V, and (ii) in Section VI.

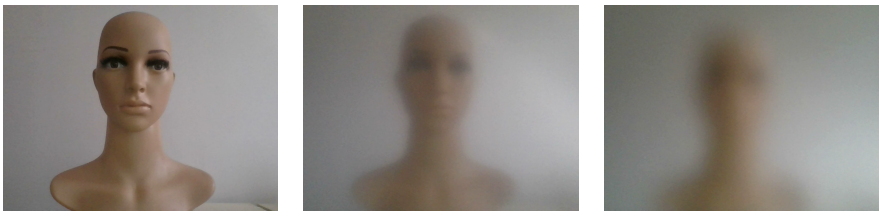
B. Application Scenario

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 be done to every new user, 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:

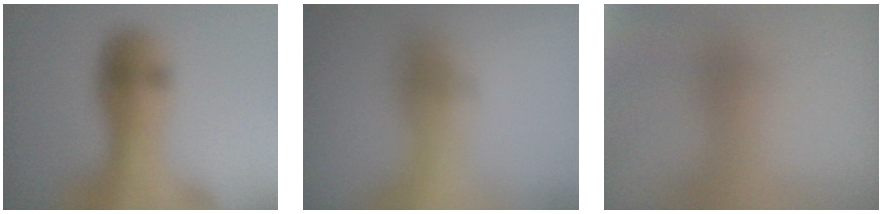
- **Scenario 1 - Same person and fixed background:** represents workstations or desktops, located in an office/home and is always used by the same person. Enrollment is possible;
- **Scenario 2 - Different people and fixed background:** represents shared workstations in fixed places (e.g., offices). 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.

IV. Material Evaluation

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



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



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

Fig. 1: Effectiveness of blurring materials considered at a distance of 30 cm.

A. 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 280854. 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 letting light to 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;

B. 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 [14], and averaged the results. Figure 1 shows pictures of Dolores taken with different blurring materials, while Figure 2 shows the quality of images for all materials and steps. A lower Niqe value indicates the image has an 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 highest quality images (*Chair* and *Antirefl*), which from visual inspection (examples are visible on Figure 1) could preserve users' privacy. The following section provides more evidence on their privacy features and discusses material selection.

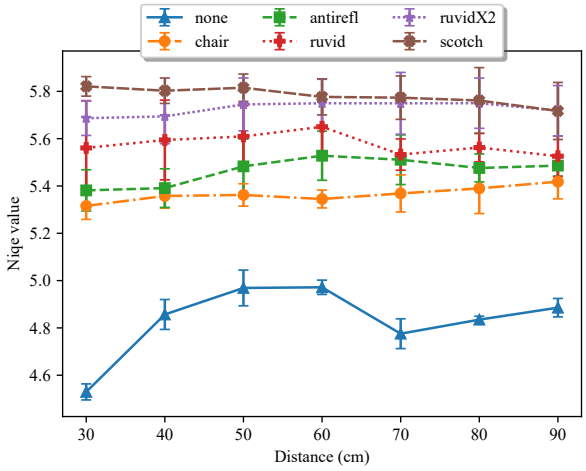


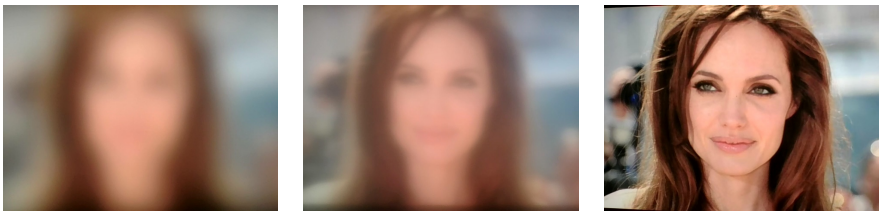
Fig. 2: Averaged quality of images for each material and steps. Lower Niqe values are associated to sharper images.

V. Material Selection

To select the best material among the two candidates from the previous section, we need to evaluate their privacy-preserving characteristics. To this extent, we first collected a

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

Fig. 3: Angelina Jolie with different blur filters.

dataset of blurred pictures of celebrities (Section V-A), and we conducted a survey asking the participants to recognize some of them (Section V-B). Last, we report the results and final decision (Section V-C).

A. 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 IV-B. To create such a dataset, we exploit the CelebA dataset [27] and the SUN dataset [44]. 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/

B. Celebrities Privacy Survey

We conducted an online survey 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 to go back and change 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 did not know the celebrity, we discarded that sample. Figure 3 shows an example of a celebrity blurred with the two filters and the original photo.

C. 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.

VI. Experiments

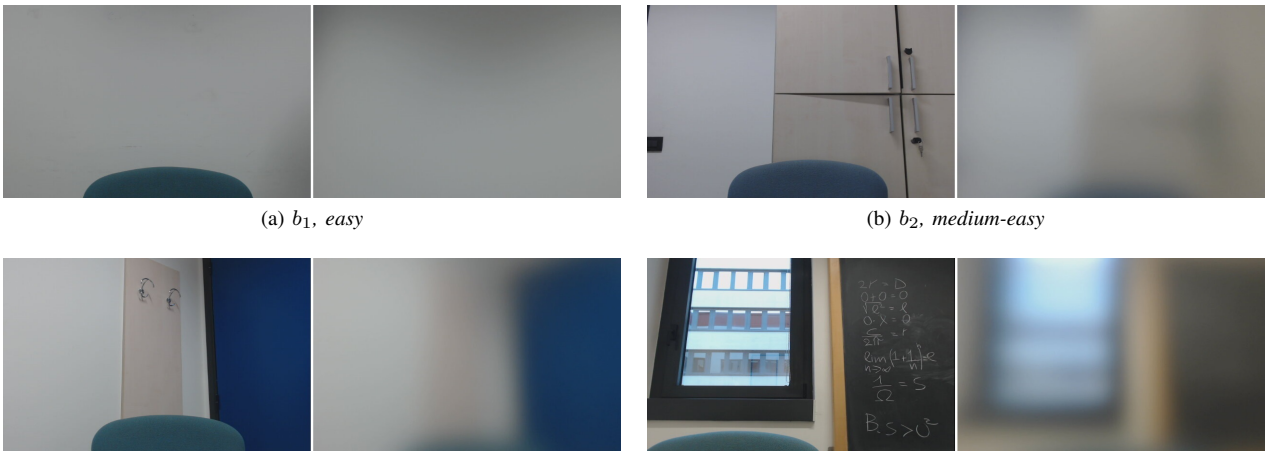
We now present the experiments we conducted to evaluate BLUFADE. In Section VI-A, we illustrate the data we collected for the experiments. Section VI-B evaluates the face detection state of the art models on our data. Last, we propose our model in Section VI-C.

A. 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 III-B, we first asked participants to follow an enrollment procedure, and then we 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 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 seconds 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 Figure 4. 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/

B. State of the Art Face Detection Algorithms

The performance of BLUFADE 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. To this extent, we extracted 240 random celebrities and 240 random enrollment images and tested with Google Cloud



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

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

Fig. 4: The four different backgrounds used in the experiments (left original, right blurred with *Antireflt*).

Vision[4], Amazon Rekognition[5], Azure Cognitive Services with detection_01 and detection_03 models[6], and TinaFace [52]. Results are reported in Table I, and they show how any of the state-of-the-art models were not suitable for our task, given the high level of blur of our images. Even Azure v3, explicitly designed for blurred faces, with 72.08% of accuracy, was not good enough for BLUFADE.

TABLE I: 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%

C. Proposed 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. [52]. Rather than binary classification (i.e., face vs. no face), we opted for object detection also 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, [43], [51] 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 [25], which uses ResNet and Feature Pyramid Network as back-bone for feature extraction. We followed an official procedure released by TensorFlow [41]. In particular, our fine-tuning procedure follows these steps: starting from ResNet pre-trained using the COCO dataset [26], 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).

1) Four Scenarios: To represent the four scenarios from Section III-B, 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 details, 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 II. In the table, we give formal and informal explanations on how we constructed the training set to understand the scenarios easily.

2) 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 adding in the training set $1,080$ celebrities' faces, and the last repetition was done fine-tuning the network using celebrities only. This way, we could see how the variance in the training set affects performance of network detection. The case of celebrities only was possible just in the fourth scenario, since it was impossible to have

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

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

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

TABLE II: 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

their enrollment or more celebrities in the same background.

3) 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. Then, we tried different thresholds ranging in [0.100,0.125 0.150, ..., 0.900], selecting the one which gave the best accuracy (i.e., threshold = 0.425). We used this threshold for the rest of the experiments.

VII. Results and Discussion

We now present the results of our experiments. Section VII-A shows the performance of the face detection task. In Section VII-B, we evaluate the performance of BLUFADE in de-authenticating people. Last, we discuss current limitation of our system in Section VII-C.

A. Face Detection

Table III 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, T3 scores the best, probably because are composed of 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 lead 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 completely 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 VII-B, we better analyze this task to implement the de-authentication system.

Looking at the scenarios, surprisingly, those without enrollment (i.e., Scenarios 2,4) 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

TABLE III: 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).

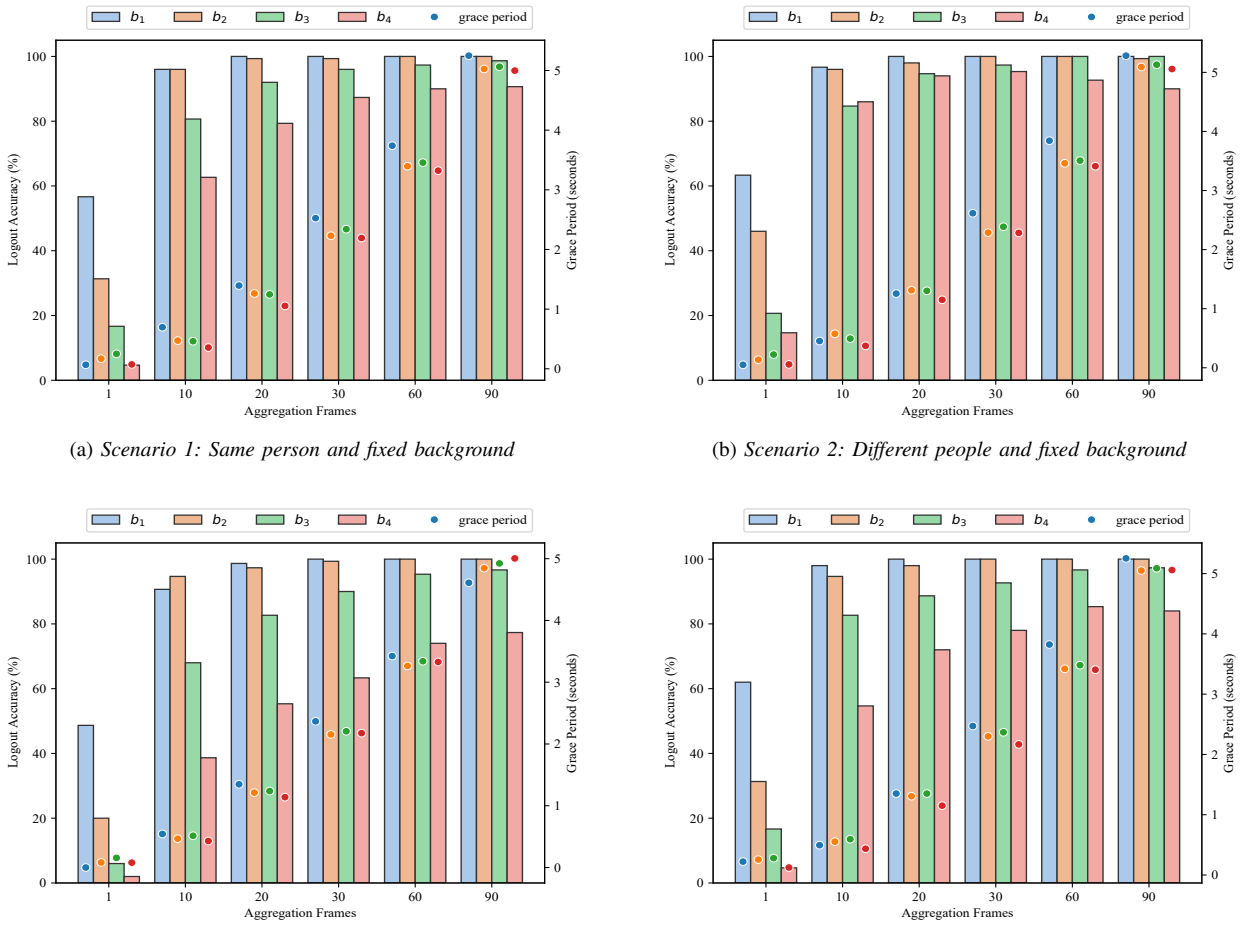
		Scenario 1					**Scenario 2**					**Scenario 3**					**Scenario 4**				
Training	*Task*	b_1	b_2	b_3	b_4	***Avg***	b_1	b_2	b_3	b_4	***Avg***	b_1	b_2	b_3	b_4	***Avg***	b_1	b_2	b_3	b_4	***Avg***
People	T1	1.00	0.99	0.95	0.86	**0.95**	1.00	0.99	0.95	0.91	**0.96**	0.99	0.99	0.93	0.80	**0.93**	0.99	0.98	0.93	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**

people in different tasks. In fact, the great differences are again in T4, T5, thus a more general network can help in such difficult 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 the 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 on 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 I), we clearly outperform them. Against Azure v3, specifically built to detect blur faces, we score around 35% and 20% more on celebrities and people respectively.

B. BLUFADE *Performance*

Face detection is the heart of BLUFADE. 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 a reliable de-authentication system. In fact, de-authenticate them every time the neural network fails the prediction is not desirable, and can negatively impact the users' experience. To improve BLUFADE, 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 be still logged in even though they already left. Obviously, this period must be short enough to not 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 good grace period is below six seconds [8].

Following these considerations, BLUFADE performs face detection and evaluates the results using a sliding window of aggregates frames. The de-authentication occurs once the face is no detect for N consecutive frames. N can be 1, which means that at the first frame the face is not detected, BLUFADE de-authenticates the user, or higher. In our experiments, we



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

tested different values of N, to a maximum of 90, which means 3 seconds (the webcam recorded at 30 FPS). Figure 5 shows the logout accuracy (i.e., the times BLUFADE correctly logs out a user) per different level 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 indicates 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, BLUFADE would wrongly de-authenticate users too frequently (i.e., over 60% on average in all the scenarios and backgrounds), making our system not usable. On the other hand, considering an higher number of aggregation frames (i.e., 90 frames) the logout accuracy rate increases up to 100% for Scenario 1 (Figure 5a) and scenario 2 (Figure 5b) in b_1 and b_2, keeping the grace period under 5 seconds. Scenario 3 (Figure 5c) shows the lowest performance of BLUFADE, 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 under five seconds.

Considering all scenarios together, the difficulty of the backgrounds highly impacts the performances. More difficult is the background, less the accuracy. Starting from 30 aggregation frames, b_1 reaches 100% of accuracy in all the scenarios, keeping the grace period below 3 seconds. b_2 shows similar performance, reaching 100% of accuracy in less than 4 seconds 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 reveals that BLUFADE 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, BLUFADE always performs above 90%.

Figure 6 compares the averaged BLUFADE 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 in the task. This is understandable since when using People only, the training and test set are more similar, which is preferable. In this case, BLUFADE achieves 96% accuracy in less than 4 seconds. On the other hand, when fine-tuning the network only using Celebrities, the training and test set are very different. Still, BLUFADE achieve almost 90% accuracy in less than 5 seconds, which is remarkable.

C. Limitations

Though BLUFADE achieves good performance, it has some limitations. First, our participants set include few ethnicity, and subjects were tested in just four backgrounds. We added more

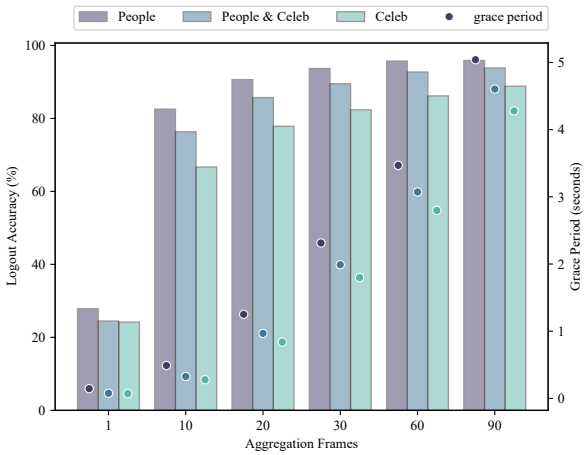


Fig. 6: Logout accuracy (bars) and grace period (dots) for different aggregation frames and training sets. The accuracy and grace periods are averaged on all the background and scenarios.

variance using the celebrities dataset, and the good results suggest BLUFADE would work even with different people. Still, more evaluations need to be conducted. Nonetheless, the four scenarios give us a good idea of how BLUFADE would work in the real world. Second, participants performed their tasks for ten seconds each. Clearly, longer use of BLUFADE needs to be evaluated. Finally, our evaluation focused on frames containing one person. Since RetinaNet can detect multiple objects in a single image, we assume it can also cope with multiple faces. This needs to be assessed.

VIII. Conclusions & Future Work

In this work, we presented BLUFADE, a de-authentication system based on blurred face detection deep learning algorithm. We conducted extensive experiments to select the physical blurring material for BLUFADE, 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. By continually detecting the users' blurred faces, BLUFADE automatically de-authenticates them with very high accuracy, i.e., up to 100% in under 3 seconds on simple backgrounds, or 96% within 4 seconds considering also difficult ones. We tested BLUFADE 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.

As future work, we plan to better assess the security of the system, testing whether is possible to reconstruct facial traits from physical blurry images. Another possible direction to expand our work is to implement a tracking system instead of performing face detection frame by frame. This way, a higher security level could be achieved without hurting usability.

REFERENCES

[1] S. Ayeswarya and J. Norman. A survey on different continuous authentication systems. *International Journal of Biometrics*, 11(1):67–99, 2019.

[2] S. P. Banerjee and D. L. Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012.

[3] S. P. Banerjee and D. L. Woodard. Biometric authentication and identification using keystroke dynamics: A survey. *Journal of Pattern Recognition Research*, 7(1):116–139, 2012.

[4] G. Borghi, M. Venturelli, R. Vezzani, and R. Cucchiara. Poseidon: Face-from-depth for driver pose estimation. In *Proceedings of the IEEE CVPR*, pages 4661–4670, 2017.

[5] M. Brocker and S. Checkoway. iseeyou: Disabling the macbook webcam indicator led. In *23rd USENIX*, pages 337–352, 2014.

[6] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE transactions on pattern analysis and machine intelligence*, 15(10):1042–1052, 1993.

[7] H. D. Company. Hp presence aware. https://tinyurl.com/HPunattended, 2020. Accessed: October, 2021.

[8] M. Conti, G. Lovisotto, I. Martinovic, and G. Tsudik. Fadewich: fast deauthentication over the wireless channel. In *2017 IEEE 37th ICDCS*, pages 2294–2301. IEEE, 2017.

[9] M. Conti and P. P. Tricomi. Pvp: Profiling versus player! exploiting gaming data for player recognition. In *International Conference on Information Security*, pages 393–408. Springer, 2020.

[10] M. Conti, P. P. Tricomi, and G. Tsudik. De-auth of the blue! transparent de-authentication using bluetooth low energy beacon. In *ESORICS*, pages 277–294. Springer, 2020.

[11] M. D. Corner and B. D. Noble. Zero-interaction authentication. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 1–11. ACM, 2002.

[12] D. Crouse, H. Han, D. Chandra, B. Barbello, and A. K. Jain. Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data. In *2015 ICB*, pages 135–142, 2015.

[13] R. Damaševičius, R. Maskeliūnas, A. Venčkauskas, and M. Woźniak. Smartphone user identity verification using gait characteristics. *Symmetry*, 8(10):100, 2016.

[14] K. De and V. Masilamani. Image sharpness measure for blurred images in frequency domain. *Procedia Engineering*, 64:149–158, 2013.

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

[16] I. Gruber, M. Hlaváč, M. Železnỳ, and A. Karpov. Facing face recognition with resnet: Round one. In *International Conference on Interactive Collaborative Robotics*, pages 67–74. Springer, 2017.

[17] G. Guo and N. Zhang. A survey on deep learning based face recognition. *Computer vision and image understanding*, 189:102805, 2019.

[18] J. Hattem. Fbi director: Cover up your webcam. https://thehill.com/policy/national-security/295933-fbi-director-cover-up-your-webcam, 2016. Accessed: September, 2021.

[19] L. Hernández-Álvarez, J. M. de Fuentes, L. González-Manzano, and L. Hernández Encinas. Privacy-preserving sensor-based continuous authentication and user profiling: a review. *Sensors*, 21(1):92, 2021.

[20] O. Huhta, P. Shrestha, S. Udar, M. Juuti, N. Saxena, and N. Asokan. Pitfalls in designing zero-effort deauthentication: Opportunistic human observation attacks. In *NDSS*, 02 2016.

[21] R. Jafri and H. R. Arabnia. A survey of face recognition techniques. *journal of information processing systems*, 5(2):41–68, 2009.

[22] T. Kaczmarek, E. Ozturk, and G. Tsudik. Assentication: User deauthentication and lunchtime attack mitigation with seated posture biometric. In *International Conference on Applied Cryptography and Network Security*, pages 616–633. Springer, 2018.

[23] T. Kanade. Picture processing system by computer complex and recognition of human faces. 1974.

[24] B.-N. Kang, Y. Kim, and D. Kim. Pairwise relational networks for face recognition. In *Proceedings of ECCV*, pages 628–645, 2018.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 2014.

[27] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of ICCV*, December 2015.

[28] X. Lu, Y. Yang, W. Zhang, Q. Wang, and Y. Wang. Face verification with multi-task and multi-scale feature fusion. *entropy*, 19(5):228, 2017.

[29] D. Machuletz, H. Sendt, S. Laube, and R. Böhme. Users protect their privacy if they can: Determinants of webcam covering behavior. In *Proceedings of EuroSEC'16*, 2016.

[30] U. Mahbub, V. M. Patel, D. Chandra, B. Barbello, and R. Chellappa. Partial face detection for continuous authentication. In *2016 IEEE ICIP*, pages 2991–2995. IEEE, 2016.

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

[32] D. Marques, I. Muslukhov, T. Guerreiro, L. Carriço, and K. Beznosov. Snooping on mobile phones: Prevalence and trends. In *Twelfth SOUPS 2016)*, 2016.

[33] I. Masi, Y. Wu, T. Hassner, and P. Natarajan. Deep face recognition: A survey. In *2018 31st SIBGRAPI*. IEEE, 2018.

[34] G. Peng, G. Zhou, D. T. Nguyen, X. Qi, Q. Yang, and S. Wang. Continuous authentication with touch behavioral biometrics and voice on wearable glasses. *IEEE transactions on human-machine systems*, 47(3):404–416, 2016.

[35] P. Perera and V. M. Patel. Face-based multiple user active authentication on mobile devices. *IEEE Transactions on Information Forensics and Security*, 14(5):1240–1250, 2018.

[36] P. J. Phillips and A. J. O'toole. Comparison of human and computer performance across face recognition experiments. *Image and Vision Computing*, 32(1):74–85, 2014.

[37] K. B. Rasmussen, M. Roeschlin, I. Martinovic, and G. Tsudik. Authentication using pulse- response biometrics. In *NDSS*, 2014.

[38] P. Samangouei, V. M. Patel, and R. Chellappa. Facial attributes for active authentication on mobile devices. *Image and Vision Computing*, 58:181–192, 2017.

[39] C. Shen, T. Yu, S. Yuan, Y. Li, and X. Guan. Performance analysis of motion-sensor behavior for user authentication on smartphones. *Sensors*, 16(3):345, 2016.

[40] S. Sinclair and S. W. Smith. Preventative directions for insider threat mitigation via access control. In *Insider Attack and Cyber Security*, pages 165–194. Springer, 2008.

[41] TensorFlow. Eager few shot object detection colab. https://tinyurl.com/FineTuningTF, 2020. Accessed: January, 2021.

[42] C. M. TEY, P. GUPTA, and D. GAO. I can be you: Questioning the use of keystroke dynamics as biometrics.(2013). In *20th NDSS 2013*, pages 1–16, 2013.

[43] I. Vasiljevic, A. Chakrabarti, and G. Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*, 2016.

[44] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

[45] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 24(1):34–58, 2002.

[46] S. Yang, P. Luo, C.-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016.

[47] S. Yang, P. Luo, C. C. Loy, and X. Tang. Faceness-net: Face detection through deep facial part responses. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1845–1859, 2017.

[48] U. Zafar, M. Ghafoor, T. Zia, G. Ahmed, A. Latif, K. R. Malik, and A. M. Sharif. Face recognition with bayesian convolutional networks for robust surveillance systems. *EURASIP Journal on Image and Video Processing*, 2019(1):1–10, 2019.

[49] S. Zafeiriou, C. Zhang, and Z. Zhang. A survey on face detection in the wild: past, present and future. *Computer Vision and Image Understanding*, 138:1–24, 2015.

[50] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM CSUR*, 35(4), 2003.

[51] Y. Zhou, D. Liu, and T. Huang. Survey of face detection on low-quality images. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 769–773, 2018.

[52] Y. Zhu, H. Cai, S. Zhang, C. Wang, and Y. Xiong. Tinaface: Strong but simple baseline for face detection. *arXiv:2011.13183*, 2020.