# COLFISPOOF: A new Database for Contactless Fingerprint Presentation Attack Detection Research

Jascha Kolberg, Jannis Priesnitz, Christian Rathgeb, and Christoph Busch
da/sec - Biometrics and Security Research Group
Hochschule Darmstadt, Germany
{firstname.lastname}@h-da.de

## Abstract

*Contactless fingerprint recognition is known for its high user comfort and low hygienic concerns. However, contactless fingerprint recognition, especially in mobile and unsupervised scenarios, is vulnerable to presentation attacks. Presentation Attack Detection (PAD) in biometric systems like contactless fingerprint recognition is more challenging compared with contact-based modalities because many detection mechanisms rely on direct contact between the finger and the surface of the capture device. Hence, in contactless scenarios it is generally possible to present more Presentation Attack Instruments (PAIs) like printout or replay artefacts.*

*In this work, we introduce COLFISPOOF, a new database for contactless fingerprint PAD. The database is acquired using a contactless fingerprint recognition system utilizing a smartphone as capturing device. It comprises 7,200 samples of 72 different PAI species and was captured with two different smartphone models. The database is publicly available for research purposes such that interested researchers can download and use it to develop new PAD algorithms.*

*Moreover, we define evaluation protocols for training and testing of machine learning algorithms such that future PAD algorithms can be benchmarked on this database in a comparable and reproducible way.*

## 1. Introduction

Contactless fingerprint recognition has become a more convenient alternative to contact-based recognition schemes [14, 17, 26]. In contrast to contact-based capturing schemes where the finger is pressed onto a planar surface, contactless fingerprint capturing systems do not require any contact between the subject and the capturing subsystem. Contactless fingerprint recognition schemes typically have a higher user acceptance, especially when multiple users interact with one single device. Here, the subjects might have fewer hygienic concerns using a contactless device [7, 15].

Contactless fingerprint capturing setups range from expensive stationary devices which capturing 3D samples to lightweight mobile setups. However, it can be observed from the literature that many contactless fingerprint recognition systems utilize mobile handheld devices like smartphones as capturing devices [17, 26]. Most contactless fingerprint recognition workflows suffer from distinct issues like an inferior biometric performance, environmental influences, or vulnerabilities to presentation attacks. Especially, mobile capturing setups based on smartphones suffer from these limiting factors because only a single type of camera is used in capturing setups and the computational capabilities are limited. In general, contactless fingerprint capturing and pre-processing are well-studied, whereas only a few works investigate the vulnerabilities of the contactless fingerprint recognition setups against presentation attacks and corresponding countermeasures. Apart from attacks which have been proven to be successful in the contact-based domain, new unknown PAIs represent a threat for contactless setups. As mentioned before, a wide variety of easy to implement PAI species can be successfully captured with a camera.

In the contact-based domain, countermeasures against attack presentations can be directly integrated into the capture device, where for example the finger's impedance is measured while touching the surface [8]. These countermeasures are not implementable in contactless biometric systems. For this reason, it is assumed that contactless systems generally exhibit a higher vulnerability to attack presentations.

### 1.1. Related Work

As mentioned, only a few works investigate contactless fingerprint presentation attack detection (PAD). Wang et al. [22] propose a prototypical capturing setup which includes a so-called aliveness testing. The authors consider the ridge-valley characteristics in the Fourier domain, re-

Table 1: Overview on databases for contactless fingerprint PAD research.

| Authors | Year | Methods | PAI species | Samples | Capturing method | available |
|---|---|---|---|---|---|---|
| Stein et al. [19] | 2013 | printout, overlay | N/A | N/A | manual capturing | no |
| Taneja et al. [20] | 2016 | printout, replay | 4 | 8,192 | manual capturing | yes |
| Wasnik et al. [23] | 2018 | printout, replay | 3 | 150 | video capturing | no |
| Our database | 2022 | printout, overlay | 72 | 7,200 | automatic capturing | yes |

flection properties, and a fingerprint quality assessment algorithm for the detection. However, the authors do not report any detection results.

Stein et al. [19] integrate a PAD algorithm in a smartphone-based contactless fingerprint capturing application. The proposed algorithm analyzes frames of a captured video with regard to reflection properties. The assumption is that PAI have a different reflection property compared to bona fide skin. In their experiments, the authors show that 77% of the attacks were successfully detected. The authors also experiment with countermeasures using a challenge response protocol. Here, the subject is asked to perform a movement, e.g. move the finger towards the capturing device. The assessment whether it is a bona fide presentation or an attack presentation is done based on the alteration between frames. It should be noted that overlays worn on top of bona fide fingertips are most likely not detected with this method.

Taneja et al. [20] present a preliminary study on fingerphoto PAD on mobile devices. Their algorithm works independent of a capturing device and is able to detect printout and replay attack presentations using a display device. It uses a Support Vector Machine (SVM) to separate bona fide presentations from attack presentations. The features are extracted using handcrafted texture descriptors. It should be noted that the algorithms are optimized for the related PAI species. The authors are able to achieve a Detection Equal Error Rate (D-EER) of 3.71%. The IIITD Spoofed Fingerphoto Database [18] used by the authors is publicly available and comprises screen-replay attacks and printout attacks. The IIITD Spoofed Fingerphoto Database is also used for a PAD approach based on a Convolutional Neural Network (CNN) by Fujio et al. [3]. The authors report a half total error rate of 0.04% for fingerphoto PAD. In addition, the authors also work on face PAD within the same publication.

Marasco and Vurity [12] trained ResNet and AlexNet CNN architectures on the IIITD Spoofed Fingerphoto Database. The CNN methods were able to achieve a D-EER of 2.14% for AlexNet and 0.97% for ResNet, respectively. In a second study, the authors evaluate different color spaces and CNN architectures and were able to slightly improve the detection accuracy compared to the baseline [13].

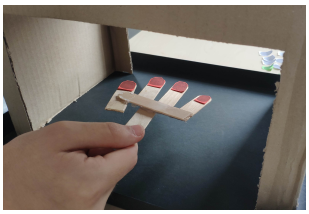
Wasnik et al. [23] present a PAD method for smartphone-based fingerprint recognition using second order local structures. The authors captured their own database, which included replay and printout attacks. Using handcrafted feature extractors in combination with an SVM, the authors achieve a D-EER of 4.43%.

From the related work it is observable that new proposals are based either on databases which are not publicly available or on the IIITD Spoofed Fingerphoto Database which includes only printout and replay attacks. Also, it should be noted, that the aforementioned database contains samples of low quality due to low contrast and sharpness. These samples are most likely not suitable for a recognition workflow because they contain no extractable features. The image quality of their attack presentations is so low that often the ridge line pattern is not visible anymore.

Moreover, many more PAI species have already proven to successfully circumvent contact-based fingerprint recognition setups, such as gelatin overlays or dragonskin [6]. These PAIs can also successfully attack contactless recognition workflows, as was shown e.g. for camera-based stationary fingerprint capture devices [9, 10]. For this reason, a database incorporating these PAIs paves the way for the development of more comprehensive and robust contactless fingerprint PAD algorithms.

## 1.2. Contribution

To overcome the aforementioned limitations, we propose COLFISPOOF, a database for contactless fingerprint PAD. The database includes 7,200 samples of 72 different PAI species and its stats are compared to previous contactless PAD databases in Table 1. In order to grant unrestricted access to our database, the ridge-line patterns of the PAIs were generated synthetically. Most notably, all attack presentations are captured with an automatic contactless fingerprint recognition workflow, which includes several quality control algorithms. This ensures that the captured samples are generally suited to be processed in an recognition workflow. Moreover, this work shows that the considered capturing workflow is also vulnerable to attacks from the contact-based domain, which have not yet been tested on contactless capturing devices. Due to the fact, that the considered capturing and recognition workflow comprises well-established methods, it is assumed that other recognition workflows are also vulnerable to these types of attacks.



(a) Printout attacks.

(b) Overlays.

Figure 1: Illustration of the capturing workflow.

Moreover, we define evaluation protocols for machine learning-based PAD algorithms such that future works based on this database remain comparable. This includes a baseline approach as well as more challenging Leave One Out (LOO) protocols to test the PAD performance in presence of unseen attacks. This is a very important factor for the development of generalizable PAD methods.

The rest of the paper is structured as follows: Section 2 describes the experimental setup considered for the database acquisition. In Section 3, the data collection is presented. The suggested evaluation protocols for machine learning algorithms are introduced in Section 4. Finally, Section 5 concludes this work.

## 2. Experimental Setup

This section describes the experimental setup we used for our database acquisition. We used an automated contactless fingerprint recognition workflow proposed in [15] to capture the attack presentations. The method is able to capture and process four fingertips at a time with one single capture attempt in order to obtain contactless fingerprint samples. The application is implemented as Android app which is able to run on any modern smartphone device. Using the in-built camera, it processes a continuous stream of images. The following pre-processing and quality control steps are applied:

1. The hand-area is segmented from the background by a color-based method. Here, the size of the bounding rectangle is tested in terms of size. A too small area indicates that the distance between hand and capturing device is too high and the sample is discarded.

2. The segmented the hand-image is split into four finger images. Again, the size of every finger image has to be in certain boundaries to ensure that the split was successful.

3. The finger images are cropped approximately to the first finger knuckle in order to include only relevant information of the fingertip.

(a) RGB image.

(b) Extracted fingerprint.

(c) Detected minutiae points.

Figure 2: The feature extraction process shown for a playdoh PAI: a) the RGB image is captured, b) the fingerprint is extracted, and c) 45 minutiae points are detected using the method of Tang et al. [21].

4. In the next step, the finger images are normalized to approximately align to a 500dpi captured contact-based fingerprint image.

5. All finger images are rotated to an upright position. Here, the rotation angle is approximated by the borders of the fingertip. Figure 2a presents the intermediate result of this stage.

6. A gray-scale transformation and a Contrast Limited Adaptive Histogram Equalization (CLAHE) is applied in order to enhance the ridge-line characteristics and to align the samples to contact-based impressions. Here, a Sobel-based sharpness metric ensures that only images with a certain level of contrast pass this quality check.

An illustrating image of the capturing method is displayed in Figure 4. A more detailed description of the recognition workflow can be found in the corresponding paper [15]. Figure 2b shows the final result of the preprocessing pipeline, which can then be used to detect minutiae points as illustrated in Figure 2c. If one of the quality checks fails, the sample is discarded and the next image is considered. The above described capturing and preprocessing workflow automatically extracts five candidate samples for every finger instance and selects the best sample with help of a quality metric. It should be noted, that no PAD algorithm is implemented into the application yet.

For our database acquisition, we used two different capturing devices, a Huawei P20 Pro and a Samsung Galaxy S9+. It should be noted that the considered app runs on most modern smartphone devices. As discussed in the original paper, both capturing systems perform equally good. The presentation attacks are captured under constrained environmental influences. A box excludes most external influences

such as illumination from the surrounding environment in order to capture high quality samples. The background of the capturing area is black, which supports the segmentation algorithm. It should be noted that the camera light is activated during the capturing process. The presentation attack is presented through openings on the side of the box and has enough space to move in all directions. This experimental setup is able to capture both, bona fide samples and presentation attacks.

For development purposes, the algorithm does not only store the final fingerprint image, but also a segmented color image of the fingerprint image (see Figure 2a). This allows researchers to train models on color images. In general, it would be possible to directly discard attack presentations during the capture process, however for a database collection it is crucial that attack presentations can be captured as well.

## 3. Data Collection

In contrast to previous works, the motivation for this data collection is to capture a high variety of different PAI species made from easily accessible materials and to enable extensive evaluations of contactless fingerprint PAD algorithms. Therefore, this dataset will be freely available for download[1].

In order to grant unrestricted access to the dataset, we decided to focus on capturing attack presentations and do not include bona fide samples, which are considered sensitive data and are subject to privacy regulations (e.g., including the right of deletion). This additionally implies that we did not capture replay attacks of printed or digital photos from bona fide fingerprints.

For bona fide samples, which are required e.g. for training and testing machine learning algorithms, we suggest using the ISPFDv1 database [18]. The database contains 4,096 samples of 128 subjects and was captured under four different environmental scenarios. The capturing device setup is comparable to the one used in this work. However, the samples in the database are not pre-processed. For this reason, we implemented a pre-processing workflow which aligns the samples of the ISPFDv1 database to our presentation attacks. The main steps of the pre-processing pipeline are segmentation of the finger area, cropping, rotation, gray scale conversion, and ridge-line enhancement. It outputs the segmented and rotated color image as well as the final enhanced sample. The pre-processing is made publicly available along with the database.

All fingerprints for the PAIs of our COLFISPOOF database are synthetically generated using synthetic fingerprint generator (SFinGe) [11] and synthetic contactless fingerprint generator (SynCoLFinGer) [16] such that the dataset does not contain fingerprints of living individuals. In recent years synthetic data has become a viable alternative. In the area of contact-based fingerprint PAD Grosz and Jain show that synthetic data significantly supports real data in detection attack presentations [4]. For iris recognition, Yadav et al. [25] propose a PAD algorithm trained on synthetic data which generalizes well on real data. Wood et al. [24] also showed that synthetic facial images are highly suitable to train machine learning algorithms for face analysis in the wild. Hence, the fact that the fingerprint patterns in our COLFISPOOF dataset are synthetically generated should not hinder the development of generalizable PAD methods.

Figure 3: Fingerprints molds created with a laser cutter.

Two research questions motivated this data collection:

1. How vulnerable are current mobile contactless fingerprint recognition systems towards simple PAIs from easily accessible materials?

2. Can we add color during the PAI casting process to obtain more threatening skin-colored PAIs?

Firstly, there are a lot of materials in their default colors available [6] that can potentially be used to attack contactless mobile fingerprint recognition systems. Since the capture process utilises a camera to take photos, every artefact that contains a fingerprint potentially threatens the system. Secondly, simple color checks might be able to detect obvious PAIs. However, transparent materials can be colored to resemble human skin. This is of particular interest due to the variety of human skin tones [1, 2] that need to be taken into account even for non-PAD scenarios. By intention, we consider PAI species of various colors in order to showcase that capturing methods are not only vulnerable to attacks colored in different skin tones.

Before we can capture the database, we need to prepare the PAIs. For that purpose, fingerprint molds were created using a laser cutter as shown in Figure 3. These molds are then filled with the casting materials from Figure 4a to transfer the fingerprint pattern onto the artefacts. Depending on the material, PAIs are directly usable (e.g., playdoh) or require some cure time to harden first (e.g., glue). Additionally, the fingerprints can also be printed on paper. Here, the black ridge-lines of SFinGe fingerprints were printed

[1]Download link: `https://dasec.h-da.de/colfispoof/`

(a) PAI Materials

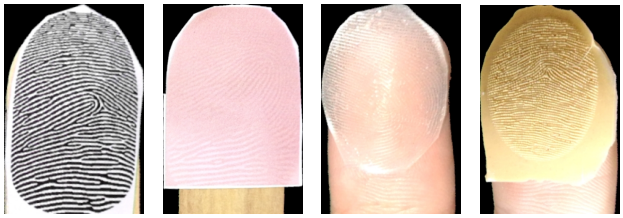

(b) Silicone colors

Figure 4: (a) Utilised materials to create fingerprint PAIs. (b) Silicone colors that were mixed with dragonskin during the casting process; from left to right: yellow, orange, brown, dark red, and red.

on white and colored sheets and the SynCoLFinGer fingerprints, which already come in different skin tones, were printed on white paper only.

Since playdoh, silly putty, and modelling clay do not harden but remain moldable by design, these artefacts can only be used for one session and need to be renewed next time. Other moldable materials such as knetosil and moldable glue will harden within a few minutes and maintain the fingerprint pattern. Then, for the group of transparent materials it is desirable to obtain thin artefacts such that the target fingerprint is visible, but additionally the skin below if worn on top of the fingertip (see Figure 5g). Finally, both dragonskin and ecoflex are transparent two part silicones. Meaning, each part by itself remains liquid but combining both parts will then harden after some time (e.g., depending on the specifications the cure time is between five minutes and 16 hours). This casting process is especially suited to add liquid silicone colors and generate PAIs that resemble human skin tones. In this context, five different colors, as shown in Figure 4b, were added separately as well as in different combinations before pouring the silicone into the molds. Since there is no visual difference between dragonskin and ecoflex, the majority of colored silicone PAIs were created with dragonskin because the cure time was more convenient (e.g., not too fast that it hardens while stirring the color but also fast enough that multiple artefact per day can be created). It should be noted that for two PAI species some leftover color remained in the molds from the previous casting process. While an attacker would most likely not create such PAIs, it might still be interesting to include these samples when evaluating (deep) machine learning algorithms. Hence, those samples were kept but *dirty* was appended to the PAI names to highlight this. Additionally, once there was not enough color to fully occlude the transparency, which was then named *brown transparent*. Example captures of different PAI species are shown in Figure 5.

(a) Knetosil (b) Mould. glue (c) Latex (d) Silly putty



(e) Paper (f) Paper (g) School glue (h) Dragonskin

Figure 5: Example captures of different PAI species. The full resolution images of all samples allow the extraction of the target fingerprint pattern.

The data collection comprises a total amount of 7,200 samples of 72 different PAI species. Here, multiple instances were created for each PAI species and 100 samples per species were captured. Since the smartphone photos are RGB images, all different colors of the same base material are considered a distinct PAI species. Especially, when mixing silicone colors, different shares of the same colors result in lighter or darker versions of previous attempts. However, this is a desired effect because human skin tones also have different levels of melanin and for the PAIs we are trying to resemble this. The full list of all PAI species of this dataset is summarised in Table 2.

So far, for all of the listed PAI species the capturing pipeline was able to extract the fingerprint pattern from captured samples. However, there were also materials that could not be captured at all. This includes black PAIs such as playdoh, moldable glue, and a 3D printed finger, where the contrast between ridge lines and valleys was not recognizable such that no fingerprint was detected at all. Similarly, casting monster latex and edible gelatin PAIs was successfully but the fingerprint pattern was not detected by the capturing process due to the structure and reflectivity of those materials. This shows that the capturing process of contactless mobile fingerprints itself may be seen as a default PAD for specific PAI species.

## 4. Evaluation Protocols

For the evaluation protocols, we propose a baseline protocol, where all PAI species are included in the training and validation sets, and more advanced Leave One Out (LOO)

Table 2: A detailed list of all captured PAI species, their base material, and the assigned leave one out group. For each PAI species, 100 samples are captured. ds = dragonskin, ef = ecoflex

| PAI | material | LOO group | PAI | material | LOO group |
|---|---|---|---|---|---|
| wood glue | glue | default color | printout blue | paper printout | printout |
| knetosil 45 | knetosil | default color | printout blue light | paper printout | printout |
| knetosil 90 | knetosil | default color | printout green | paper printout | printout |
| latex fashion flesh | latex | default color | printout green light | paper printout | printout |
| modelling clay | modelling clay | default color | printout orange | paper printout | printout |
| moldable glue blue | moldable glue | default color | printout orange light | paper printout | printout |
| moldable glue brown | moldable glue | default color | printout red | paper printout | printout |
| moldable glue green | moldable glue | default color | printout rose | paper printout | printout |
| moldable glue grey | moldable glue | default color | printout white | paper printout | printout |
| moldable glue orange | moldable glue | default color | printout yellow | paper printout | printout |
| moldable glue pink | moldable glue | default color | printout yellow light | paper printout | printout |
| moldable glue red | moldable glue | default color | printout syncolfinger | paper printout | printout |
| moldable glue white | moldable glue | default color | | | |
| moldable glue yellow | moldable glue | default color | ds original | dragonskin | transparent |
| playdoh blue | playdoh | default color | ds brown transparent | dragonskin | transparent |
| playdoh blue light | playdoh | default color | gelafix | gelafix | transparent |
| playdoh brown dark | playdoh | default color | gelatin fx | gelatin | transparent |
| playdoh brown light | playdoh | default color | school glue | glue | transparent |
| playdoh green dark | playdoh | default color | | | |
| playdoh green light | playdoh | default color | ds brown | dragonskin | colored silicone |
| playdoh orange | playdoh | default color | ds brown darkred | dragonskin | colored silicone |
| playdoh orange neon | playdoh | default color | ds darkred | dragonskin | colored silicone |
| playdoh pink | playdoh | default color | ds darkred brown | dragonskin | colored silicone |
| playdoh pink pale | playdoh | default color | ds darkred brown yellow | dragonskin | colored silicone |
| playdoh purple | playdoh | default color | ds orange | dragonskin | colored silicone |
| playdoh purple dark | playdoh | default color | ds orange brown | dragonskin | colored silicone |
| playdoh red | playdoh | default color | ds orange brown dark | dragonskin | colored silicone |
| playdoh teal | playdoh | default color | ds orange brown darkred | dragonskin | colored silicone |
| playdoh white | playdoh | default color | ds orange brown light | dragonskin | colored silicone |
| playdoh yellow | playdoh | default color | ds orange dirty | dragonskin | colored silicone |
| playdoh yellow light | playdoh | default color | ds red | dragonskin | colored silicone |
| sillyputty gold | silly putty | default color | ds red brown | dragonskin | colored silicone |
| sillyputty green | silly putty | default color | ds yellow | dragonskin | colored silicone |
| sillyputty pink | silly putty | default color | ds yellow brown | dragonskin | colored silicone |
| sillyputty red | silly putty | default color | ef brown yellow darkred | ecoflex | colored silicone |
| sillyputty silver | silly putty | default color | ef brown yellow darkred orange | ecoflex | colored silicone |
| sillyputty yellow | silly putty | default color | ef yellow brown dirty | ecoflex | colored silicone |

protocols to analyze the PAD performance in presence of unknown attacks.

The baseline scenario randomly splits the samples for each PAI species into training (30%), validation (20%), and test (50%) partitions. These non-overlapping partitions ensure that PAD algorithms are tested on data unseen during training and validation and thus guarantee a fair evaluation. With 100 samples per PAI species, this results in 2,160 samples in the training partition, 1,440 samples for validation, and 3,600 samples for testing. In order to have an unbiased training result for two-class classifiers, we recommend the same number of bona fide samples for the training and validation partitions. For PAD it is only relevant whether the presentation is bona fide or an attack and not e.g. which material was presented. Additionally, all samples of a bona fide subject should only occur in either training, validation, or test partition due to their natural similarity.

The idea of LOO protocols is that selected PAI species are not seen during training, including validation, and are only available for testing. On the other hand, the test set does not include other PAI species but the ones left out from training for easy result analysis. This setting allows to evaluate the generalizability of the PAD methods regarding unknown attacks. However, with 72 different PAI species it does not make sense to leave only one PAI species out at a time. The results would not indicate the generaliza-

Table 3: Number of PAI samples for each partition for the different evaluation protocols.

| protocol | train | validation | test |
|---|---|---|---|
| baseline | 2,160 | 1,440 | 3,600 |
| LOO printout | 4,200 | 1,800 | 1,200 |
| LOO transparent | 4,690 | 2,010 | 500 |
| LOO default color | 2,450 | 1,050 | 3,700 |
| LOO colored silicone | 3,780 | 1,620 | 1,800 |

tion capabilities anymore but rather how similar the left out PAI species is to already seen ones. Hence, four separate LOO groups are created, which comprise similar PAI species such that a full LOO group is excluded from training. These groups are defined as *i)* printout PAIs, *ii)* transparent PAIs, *iii)* PAIs in their defeault color, and *iv)* manually colored silicone PAIs, where liquid color was added during the casting process. The assignment of each PAI species to those groups is included in Table 2. Based on this partitioning, it is now possible to analyze the vulnerability towards different attack scenarios. In this context, the printout group requires the least expertise from the attacker. For the default colored PAIs, the attacker just needs to acquire the material and cast the PAI using a mold with the target fingerprint pattern. The same holds for the last two groups, but their appearance translates to more advanced attack scenarios: while transparent PAIs worn on top of the attackers fingers still show the natural skin tone underneath, it is also possible to add colors during the casting process and obtain PAIs in different skin tones or at least colors that are close to natural skin tones.

For the protocols, the LOO group is solely present in the test partition and all other samples of each PAI species are split into train (70%) and validation (30%) sets. The different protocols are summarized in Table 3 in terms of samples per partition. The exact mapping of samples and partitions for all protocols can be downloaded together with the dataset.

Since this database does not include bona fide samples to allow unrestricted access to the data, those samples need to be added before training a PAD algorithm. However, since the background or image size might differ, PAD algorithms might focus on those differences to *learn* the distinction between bona fide presentations and attack presentations. Hence, we suggest to extract a Region of Interest (ROI) from within the fingertip, thus excluding unwanted deviations. Python code to extract a fingertip ROI of $100 \times 200$ pixels, which also takes into account slightly tilted fingers, is included in the supplementary material of the database. The code works both for attack samples and bona fide samples. This ROI cropping enables PAD algorithms to analyze the real differences between bona fide presentations and attack presentations. Furthermore, state of the art deep learning models mostly require a fixed input size of the images, which is by default not given in contactless mobile capture scenarios since the fingers' distance to the camera can vary for multiple presentations.

In general, mobile contactless captures include unconstrained environment depending on the surroundings. Thus, if we want to develop PAD algorithms that generalize across different capture devices and capture locations (i.e., indoor and outdoor), we need to exclude all external factors from the PAD algorithm and focus on a central fingertip ROI.

Finally, results should be reported according to the standardized metrics defined in ISO/IEC 30107-3 [5]:

- Attack Presentation Classification Error Rate (APCER): proportion of attack presentations wrongly classified as bona fide presentations

- Bona fide Presentation Classification Error Rate (BPCER): proportion of bona fide presentations wrongly classified as attack presentations

For related works it additionally has proven useful to fix e.g. BPCER = 0.2% and report the corresponding APCER values in order to benchmark different PAD algorithms for a specific operation point. This approach suits the overall goal that the system remains user friendly (i.e., bona fide presentations are not wrongly rejected), while detecting as many attack presentations as possible.

On the other hand, the Impostor Attack Presentation Accept Rate (IAPAR) is not of high relevance here since all PAIs contain synthetic fingerprints. Furthermore, this metric depends on the verification threshold of the fingerprint recognition system and thus it is not comparable across works which utilise own bona fide samples. However, it was tested that all 72 PAI species can be captured in sufficient quality to extract the fingerprint and locate minutiae points. Hence, the Attack Presentation Acquisition Rate (APAR) without any preceding PAD is at 100% for these 72 PAI species. As reported in the previous section, there were also materials that could not be captured (i.e., black playdoh), which translates to an APAR of 0%.

Generally, it should be noted that attackers need only one successful attempt from all available (including unknown) PAI species. On the other hand, research requires extensive datasets to develop and train new PAD methods. Hence, research databases are based on mass production and mass capturing of PAIs, which might result in a decrease of quality. The important point is to show that specific materials are suited to create PAIs of sufficient quality that allow feature extraction. If this is the case, the attack is a realistic threat to the system even though some samples of the dataset possibly are of less quality.

## 5. Conclusions

PAD is an important research area in the field of contactless fingerprint recognition in order to increase the security of the system. Like in other contactless biometric systems, hardware-based PAD methods which analyze the skin surface directly are challenging to implement, which makes software-based PAD algorithms necessary. Moreover, additional PAI species such as printout and replay attacks can successfully attack contactless recognition workflows. From the literature it is observable that only a few works on contactless fingerprint PAD are proposed so far and that up to now only a subset of possible attacks is investigated. One reason for this is that there is no database publicly available which includes various different PAI species.

To tackle these shortcomings, we present the publicly available COLFISPOOF database, which comprises 7,200 attack presentations from 72 different PAI species. The database is captured using an automated contactless fingerprint recognition workflow, which ensures that the samples are in general suitable for a recognition process.

To complement our contribution, we additionally provide evaluation protocols to analyze the PAD performance in known and unknown attack scenarios. Following these protocols, future works remain comparable while reporting standardized metrics.

This work paves the way for more elaborated PAD research on contactless fingerprint recognition. As next steps, countermeasures against the proposed PAIs should be investigated and PAD approaches shall be developed. Here, the proposed evaluation protocols aid the result analysis and thus the development of strong PAD algorithms. Finally, lightweight PAD methods could be directly implemented into the recognition workflow, while heavy deep learning methods should be applied after the capturing process. The overall goal is to achieve a robust and reliable PAD for mobile contactless fingerprint recognition.

## Acknowledgement

This research work has been partially funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. Additional thanks goes to Ralph Breithaupt from BSI, who kindly provided the laser-cut fingerprint molds.

## References

[1] Aeddon Berti, Nasser Nasrabadi, and Jeremy Dawson. Investigating the impact of demographic factors on contactless fingerprint interoperability. In *International Conference of the Biometric Special Interest Group (BIOSIG)*, LNI, pages 1–8. GI, September 2022.

[2] Thomas B. Fitzpatrick. The validity and practicality of sun-reactive skin types I through VI. *Archives of Dermatology*, 124(6):869–871, June 1988.

[3] Masakazu Fujio, Yosuke Kaga, Takao Murakami, Tetsushi Ohki, and Kenta Takahashi. Face/fingerphoto spoof detection under noisy conditions by using deep convolutional neural network. In *BIOSIGNALS*, pages 54–62, 2018.

[4] Steven A. Grosz and Anil K. Jain. SpoofGAN: Synthetic fingerprint spoof images. *arXiv preprint arXiv:2204.06498*, 2022.

[5] ISO/IEC JTC1 SC37 Biometrics. *ISO/IEC 30107-3. Information Technology - Biometric presentation attack detection - Part 3: Testing and Reporting*. International Organization for Standardization, 2017.

[6] Ondřej Kanich, Martin Drahanskỳ, and Martin Mézl. Use of creative materials for fingerprint spoofs. In *International Workshop on Biometrics and Forensics (IWBF)*, pages 1–8, 2018.

[7] Christof Kauba, Dominik Söllinger, Simon Kirchgasser, Axel Weissenfeld, Gustavo Fernández Domínguez, Bernhard Strobl, and Andreas Uhl. Towards using police officers' business smartphones for contactless fingerprint acquisition and enabling fingerprint comparison against contact-based datasets. *Sensors*, 21(7):2248, 2021.

[8] Jascha Kolberg, Daniel Gläsner, Ralph Breithaupt, Marta Gomez-Barrero, Jörg Reinhold, Arndt von Twickel, and Christoph Busch. On the effectiveness of impedance-based fingerprint presentation attack detection. *Sensors*, 21(17), 2021.

[9] Jascha Kolberg, Marta Gomez-Barrero, and Christoph Busch. On the generalisation capabilities of fingerprint presentation attack detection methods in the short wave infrared domain. *IET Biometrics*, 10(4):359–373, 2021.

[10] Jascha Kolberg, Marcel Grimmer, Marta Gomez-Barrero, and Christoph Busch. Anomaly detection with convolutional autoencoders for fingerprint presentation attack detection. *Transactions on Biometrics, Behavior, and Identity Science (TBIOM)*, 3(2):190–202, April 2021.

[11] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar. *Synthetic Fingerprint Generation*, pages 271–302. Springer London, 2009.

[12] Emanuela Marasco and Anudeep Vurity. Fingerphoto presentation attack detection: Generalization in smartphones. In *IEEE International Conference on Big Data (Big Data)*, pages 4518–4523, 2021.

[13] Emanuela Marasco, Anudeep Vurity, and Asem Otham. Deep color spaces for fingerphoto presentation attack detection in mobile devices. In *Computer Vision and Image Processing*, pages 351–362, Cham, 2022. Springer International Publishing.

[14] Giuseppe Parziale and Yi Chen. Advanced technologies for touchless fingerprint recognition. In *Handbook of Remote Biometrics*, pages 83–109. Springer, 2009.

[15] Jannis Priesnitz, Rolf Huesmann, Christian Rathgeb, Nicolas Buchmann, and Christoph Busch. Mobile contactless fingerprint recognition: Implementation, performance and usability aspects. *MDPI Intelligent Sensors*, 2022.

[16] Jannis Priesnitz, Christian Rathgeb, Nicolas Buchmann, and Christoph Busch. SynCoLFinGer: Synthetic contactless fingerprint generator. *Pattern Recognition Letters*, 157:127–134, 2022.

[17] Jannis Priesnitz, Christian Rathgeb, Nicolas Buchmann, Christoph Busch, and Marian Margraf. An overview of touchless 2D fingerprint recognition. *EURASIP Journal on Image and Video Processing*, 2021(1):1–28, 2021.

[18] Anush Sankaran, Aakarsh Malhotra, Apoorva Mittal, Mayank Vatsa, and Richa Singh. On smartphone camera based fingerphoto authentication. In *IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–7. IEEE, 2015.

[19] Chris Stein, Vincent Bouatou, and Christoph Busch. Video-based fingerphoto recognition with anti-spoofing techniques with smartphone cameras. In *International Conference of the Biometric Special Interest Group (BIOSIG)*, pages 1–12, 2013.

[20] Archit Taneja, Aakriti Tayal, Aakarsh Malhorta, Anush Sankaran, Mayank Vatsa, and Rieha Singh. Fingerphoto spoofing in mobile devices: a preliminary study. In *IEEE International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–7. IEEE, 2016.

[21] Yao Tang, Fei Gao, Jufu Feng, and Yuhang Liu. FingerNet: An unified deep network for fingerprint minutiae extraction. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 108–116. IEEE, 2017.

[22] Lirong Wang, Rania H. Abd El-Maksoud, Jose M. Sasian, William P. Kuhn, Kathleen Gee, and Valorie S. Valencia. A novel contactless aliveness-testing (CAT) fingerprint sensor. In R. John Koshel and G. Groot Gregory, editors, *Novel Optical Systems Design and Optimization XII*, volume 7429, page 742915. International Society for Optics and Photonics, SPIE, 2009.

[23] Pankaj Wasnik, Raghavendra Ramachandra, Kiran Raja, and Christoph Busch. Presentation attack detection for smartphone based fingerphoto recognition using second order local structures. In *International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 241–246. IEEE, 2018.

[24] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J. Cashman, and Jamie Shotton. Fake it till you make it: Face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3681–3691, October 2021.

[25] Shivangi Yadav, Cunjian Chen, and Arun Ross. Synthesizing iris images using RaSGAN with application in presentation attack detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2019.

[26] Xuefei Yin, Yanming Zhu, and Jiankun Hu. A survey on 2D and 3D contactless fingerprint biometrics: A taxonomy, review, and future directions. *IEEE Open Journal of the Computer Society*, 2:370–381, 2021.