Fingerphoto Spoofing in Mobile Devices: A Preliminary Study

Archit Taneja, Aakriti Tayal, Aakarsh Malhotra, Anush Sankaran, Mayank Vatsa, Richa Singh
IIIT Delhi, India
{archit12024, aakriti12001, aakarshm, anushs, mayank, rsingh}@iiitd.ac.in

Abstract

Biometric-based authentication for smart handheld devices promises to provide a reliable and alternate security mechanism compared to traditional methods such as pins, patterns, and passwords. Although fingerprints are a viable source for authentication, they generally require installation of an additional hardware such as optical and swipe sensors on mobile devices, and are only available in expensive, high-end smartphones. Alternatively, fingerphoto images captured using the smartphone camera for authentication is one of the promising biometric approaches. However, using fingerphotos for authentication brings along a major challenge of fingerphoto spoofing. This research is aimed at understanding the effect of spoofing on fingerphotos. There are three major contributions of this research: (i) create a large spoofed fingerphoto database and make it publicly available for research, (ii) to establish the effect of print attack and photo attack in fingerphoto spoofing, and (iii) understand the performance of existing spoofing detection algorithms on fingerphoto spoofing.

1. Introduction

With the increasing usage of handheld smart devices, users have started saving personal data and using multiple confidential services. For instance, many banking applications use multiple user authentication techniques such as pins, patterns, and passwords which happen to be inconvenient and are highly susceptible to over-the-shoulder surfing attacks. Incorporating biometrics as a method for authentication is a suitable alternative. Various smartphones such as Apple iPhone 5s, Nexus 5, and OnePlusTwo use fingerprints for unlocking the device. However, this requires an additional hardware component or an optical scanner which is only present in expensive high-end smartphones. In addition to that, there have been many instances where unauthorized users have been able to access confidential and secure documents by successfully spoofing the biometric sample [5]. For example, in 2013, German Hackers group Chaos Computer Club spoofed the fingerprint scanner in Apple iPhone 5s by using a counterfeited fingerprint [1]. A team from University of Hanoi (Vietnam) demonstrated the trick to spoof face recognition tool of Lenovo, Asus, and Toshiba laptops by using photographs of a genuine user [2].

Figure 1: Sample images illustrating the contrast between (a) a fingerphoto and (b) a fingerprint.

Using fingerphoto images captured using a mobile camera, as shown in Figure 1, an alternative biometric approach provides a potential option for user authentication in smartphone devices [14]. Applications of fingerphoto biometrics can be beyond unlocking mobile devices, such as banking and law enforcement applications[1]. However, capturing fingerphoto images is also susceptible to spoofing attacks and hence, it becomes an important research problem to study. Although it is established in the literature, mentioned in Table 1, that fingerprint based images can be spoofed, there is a lack of experimental study on spoofing using fingerphoto images. In addition to that, there is no publicly available database having spoofed and original images of the same finger, captured using a smartphone camera. Therefore, there are three major contributions in this research:

1. We experimentally study different kinds of spoofing

[1] http://goo.gl/x2wjVO

Research	Modality	Spoofed Using	Algorithm	Database	Results
Pan *et al*. [9]	Face	Replay face videos	Measuring eye-blink behavior using adaptive boosting	ZJU Eyeblink	95.7% GAR@0.1%FAR
Ohana *et al*. [12]	Fingerprint	Ink, Mikrosil mold, Gelatin	Device Dongle & RFID Middleware	Non-public fingerprint database	No accuracy reported
Akhtar *et al*. [4]	Face, Fingerprint	Silicon, Latex molds, Photo, print attack	Train a serially fused multi-biometric system	LivDet2011, Photo attack, Print attack	Photo attack: 62% FAR@EER, print attack: 40% FAR@EER
Akhtar *et al*. [5]	Face, Fingerprint, Iris	Silicon mold, Photo, Print attack	LUCID, CENTRIST, POEM features based detection	Print attack, NUAA, ATVS-FIr, ATVS-FFp	HTER - Iris: 0.01%, Face: 0.1%, Fingerprint: 0.25%
Stein *et al*. [16]	Fingerphoto	Silicon, Gelatin molds, print, photo attack	Measure light reflection by adaptive threshold	In-house non-public fingerphoto database	$1.2 - 3\%$ EER

Table 1: A literature survey of existing biometric spoofing detection algorithms in mobile devices.

attacks, such as print attack and photo attack, on fingerphotos to establish the extent to which fingerphotos can be spoofed.

2. We evaluate the performance of different features such as Local Binary Patterns (LBP), Dense Scale Invariant Feature Transform (DSIFT), and Locally Uniform Comparison Image Descriptor (LUCID) features along with Support Vector Machine (SVM) based fingerphoto spoofing detection algorithm to distinguish between spoofed and non-spoofed images.

3. We extend the IIITD SmartPhone Fingerphoto database [14] to create the *Spoofed Fingerphoto database* [2] with six different photo attack mechanisms and two different print attack mechanisms. This database is made publicly available for research.

2. Spoofed Fingerphoto Database

Since fingerphoto detection is a comparatively newer paradigm for biometric authentication, fingerphoto spoofing has not been well explored in the literature. As there is no publicly available spoofed fingerphoto database, we created one using the existing IIITD FingerPhoto Database [14]. As shown in Figure 3, the setup used to create spoofed fingerphoto database has a display mechanism and a capture mechanism kept apart at a fixed distance. The spoofed database was collected under indoor controlled illumination environment. Typically, the display device/mechanism is the system that is being spoofed while the capture device/mechanism is the spoofing device/mechanism that is used in place of the actual biometric modality. In the literature, most of the spoofing attempts have been made with state-of-the-art high resolution phone cameras. However, in this research, we also wanted to study the effect of camera variation, therefore, we utilized two different resolution cameras as capture mechanisms: (1) OnePlusOne (OPO): 13MP camera with High Dynamic Range (HDR) imaging, Flash-OFF mode, and Auto-focus and (2) Nokia C5: 3.15MP camera with Flash-OFF mode which has no Auto-focus feature.

With these two devices, we have captured two kinds of attacks: print attack and photo attack. Considering different modalities of photo attack, three types of display mechanisms are used to capture the spoofed samples.

- Print Attack: A colored paper-printout is placed in front of the capture device/mechanism to spoof the original fingerphoto. Colored images are printed with HP Color-LaserJet CP2020 Series PCL6 printer at 600 ppi.
- Photo Attack: The original image is displayed on another display device in front of the capture mechanism. Three different display mechanisms used for spoofing are: (1) Apple iPad with Retina Display with 2048×1536 resolution (2) Nexus 4 with 1280×768 resolution, and (3) Dell Inspiron N5110 Laptop with 1280×720 resolution.

This results in two different capture mechanisms and four different display mechanisms resulting in a total of eight different ways of spoofing. Figure 3 illustrates the setup for spoofing attempts. Original non-spoofed images are obtained from the IIITD SmartPhone FingerPhoto Database [14] which has two sets of fingerphotos, containing a total of 4096 images. There are 128 classes $\times$ 4 variations $\times$ 8 instances in the original dataset. The spoofed

[2] http://iab-rubric.org/resources/sfd.html

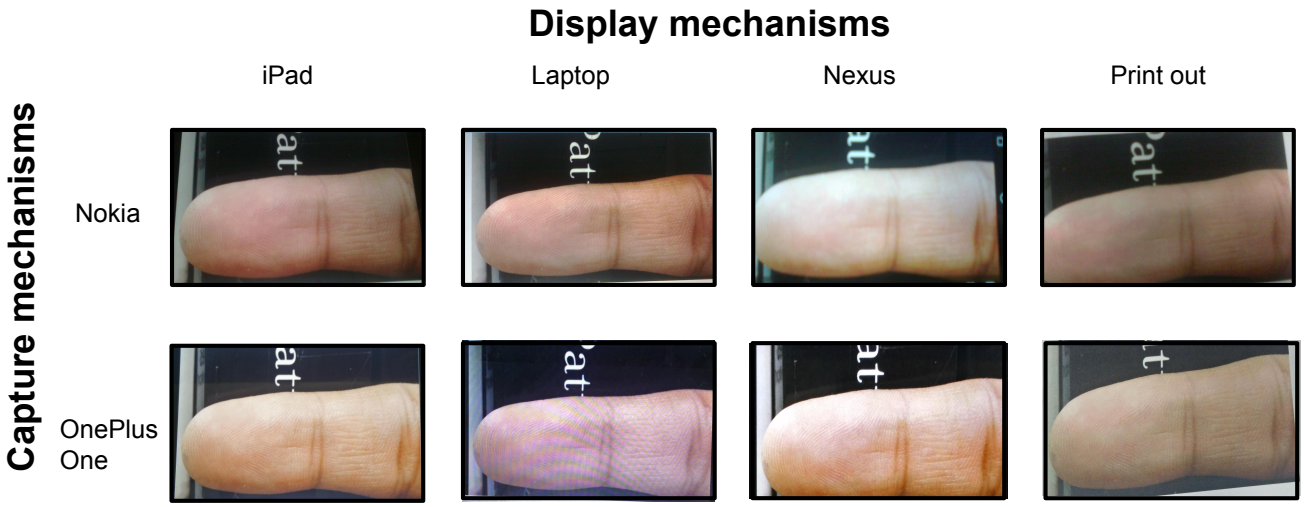


Figure 2: Eight different types of spoofing mechanisms used to create the proposed database, having two different display mechanisms and four different capture mechanisms.

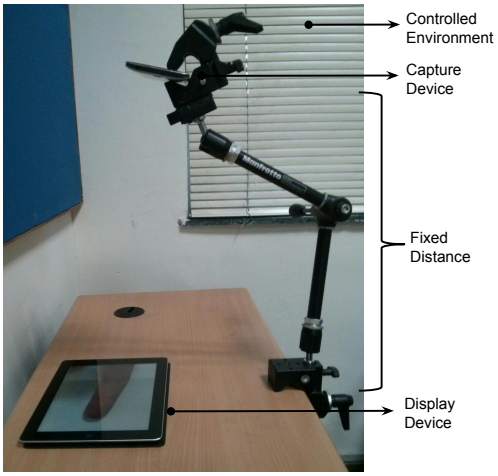


Figure 3: Camera apparatus and environmental setup used to create the spoofed fingerphoto database.

Spoof/Non-spoof	Display	Capture	# Images	Total
Print attack	Color print	Nokia	1024	2048
		OPO	1024	
Photo attack	iPad	Nokia	1024	6144
		OPO	1024	
	Nexus	Nokia	1024	
		OPO	1024	
	Laptop	Nokia	1024	
		OPO	1024	
Non-spoofed	4096			4096

Table 2: A summary of the subsets in the Spoofed Fingerphoto database.

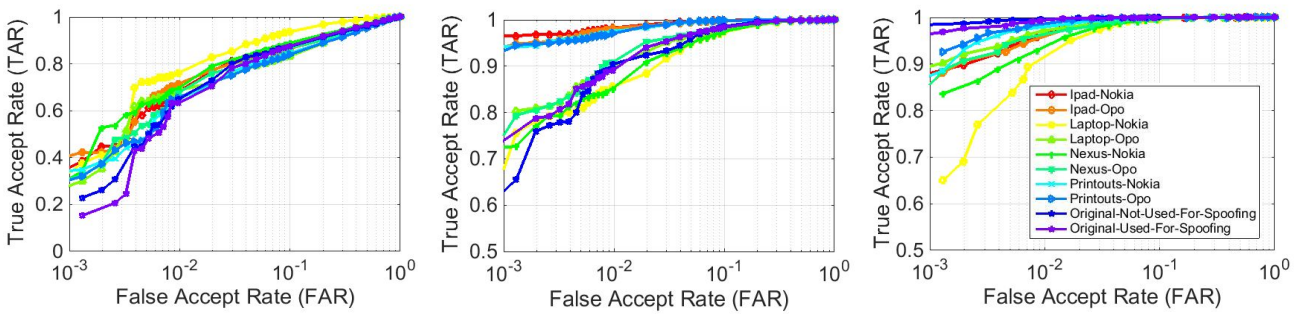
database is created by choosing 2 out of the 8 instances and spoofing it in 8 different ways. Thus, there are in total 128 classes $\times$ 4 variations $\times$ 2 instances $\times$ 2 capture mechanisms $\times$ 4 display mechanisms (1 print + 3 photo) = *8192 spoofed photos*. Table 2 summarizes the characteristics of the created database and Figure 2 shows sample fingerphoto images from the collected dataset. The proposed database has certain non-ideal constraints such as the effect of different camera resolutions, inter-sensor variations, different display mechanisms, and varying capture mechanisms, that can affect the performance of spoofing. For example, we use Nokia C5 which has a 3.15MP camera as well as OPO which has 13MP.

3. Can Fingerphoto Images be Spoofed?

The primary objective of this section is to address the question, 'Can fingerphoto images be spoofed?' and to study the extent to which a fingerphoto matching system can be spoofed. The ideal environment is when a fingerphoto matching system provides high matching performance for original images, while providing a poor matching performance for spoofed input images. Thereby, ensuring that the matching system cannot be spoofed.

3.1. Fingerphoto Matching Algorithm

For matching fingerphoto images captured using a smartphone camera, we use a state-of-the-art fingerphoto approach recently proposed by Sankaran et al. [14]. As explained in [14], the algorithm involves applying two levels of segmentation with a coarse cropping followed by the ROI extraction using skin color segmentation in CMYK channel. The segmented image is enhanced using median filtering, histogram equalization, and sharpened by subtracting the original image with the Gaussian blurred image. Scattering Network (ScatNet) based features are used as a robust rep-



(a) ScatNet + L2 (b) ScatNet + NN (c) ScatNet + RDF

Figure 4: Receiver operating characteristic (ROC) curves for different spoofing and non-spoofing subsets establishing the effectiveness of different spoof attacks.

			Equal Error Rate (%)		
Spoof Attack	Display	Capture	ScatNet + L2	ScatNet + NN	ScatNet + RDF
Print Attack	Color print	Nokia	11.18 ± 0.35	1.02 ± 0.46	1.37 ± 0.66
		OPO	12.55 ± 3.37	1.30 ± 0.43	1.63 ± 0.54
Photo Attack	iPad	Nokia	7.66 ± 1.35	4.71 ± 0.48	2.53 ± 1.08
		OPO	14.11 ± 1.88	3.65 ± 0.93	1.52 ± 0.66
	Nexus	Nokia	10.41 ± 1.09	3.68 ± 0.86	1.94 ± 0.48
		OPO	11.47 ± 0.46	3.31 ± 0.28	1.41 ± 0.38
	Laptop	Nokia	13.54 ± 2.76	1.83 ± 0.12	0.96 ± 0.11
		OPO	13.70 ± 2.15	1.50 ± 0.28	0.86 ± 0.10
Original images not used for spoofing			11.46 ± 1.53	4.23 ± 0.34	0.48 ± 0.22
Original images used for spoofing			11.43 ± 1.77	3.58 ± 0.62	0.80 ± 0.15

Table 3: The performance of different matching algorithms across the spoofed and non-spoofed subsets, captured in terms of EER % (mean $\pm$ standard deviation across three times cross validation).

resentation for fingerphoto images. ScatNet is a sequential application of wavelet transform in the image which gives a representation that is stable to local geometric transformation. It has been shown that the ScatNet features are good for extracting texture patterns in images [15]. However, as ScatNet produces high dimensional features, PCA based dimensionality reduction is performed by preserving 99% energy. Three different matching approaches are adopted: (i) L2 distance based matching, (ii) Neural Network (NN), and (iii) Random Decision Forest (RDF) [6].

3.2. Experimental Protocol

The aim of this first experiment is to determine whether fingerphoto images can be spoofed across different spoofing methods. The ScatNet based matching algorithm is trained using the IIITD SmartPhone Fingerphoto Database [14]. The database has 128 classes and it is split into non-overlapping 50% training and 50% testing sets. The training data consists of non-spoofed images in both gallery and probe sets, since it is assumed that the matching algorithm is trained using non-spoofed images only.

The training data is split into gallery and probe with (64 classes $\times$ 4 variations $\times$ 4 instances =) 1024 photos in each subset. In the IIITD SmartPhone Fingerphoto Database, each class has 8 instances with each of the 4 variations corresponding to indoor, outdoor and two background variations. 2 instances out of these 8 are selected to generate the spoofed images. Each of the 2 instances is spoofed using 8 different mechanisms. The test gallery has 2 instances that are not used for spoofing so total images are (64 classes $\times$ 4 variations $\times$ 2 instances =) 512. The test probe set has 10 different subsets: (i) 8 subsets corresponding to the 8 spoofing mechanisms, each having 512 images, (ii) 2 instances that are neither used in gallery nor used for probe are used to create a probe subset of 512 images, and (iii) 2 instances that are used for creating spoofed images are added as the last probe subset having 512 images. Thus, the first eight subsets show accuracy on spoofing database and the last two subsets match non-spoofed images to provide a baseline accuracy. To avoid the training bias, three times

random cross validation is performed on the overall train-test split and the average equal error rate (EER) along with the standard deviation is calculated.

3.3. Establishing Fingerphoto Spoofing

To establish the effect of fingerprint spoofing, ScatNet based fingerphoto matching algorithm is applied using the experimental protocol described in the previous subsection. A detailed study on the comparative analysis on using ScatNet based fingerphoto matching algorithm has been studied by Sankaran et al. [14]. The results of different matching algorithms on individual spoofed and non-spoofed subsets are summarized in Table 3 and the ROC curves are shown in Figure 4. As each of the test probe subset contains an equal number of images per class and is compared against the same test gallery, the results can be directly compared. Further, comparing the performance of different probe subsets across the same algorithm suggests about the effectiveness of the display and capture mechanisms used to create the spoofing database. Analyzing the results across multiple surfaces yeids the following observations.

- With different kinds of spoofing attacks and non-spoofed images, the equal error rate across all three matching algorithms does not vary much. ScatNet + RDF yields the best results for both spoofed and non-spoofed images; the EERs are in the range of 0.48% to 2.53%. This indicates that the matching algorithm is prone to spoofing.

- Both capture and display devices play an important role in spoofing, changing either can significantly change the matching performance. For instance, with iPad as the display device, spoofed prints collected using Nokia device yield 2.53% EER whereas changing the capturing unit to OPO reduces the error rate by 1.01%. However, it can be observed that irrespective of the resolution of the camera used for capture, using a laptop to display an image is probably the most effective method for spoofing the system. With laptops, the accuracies are close to non-spoofed query images.

- On the contrary, using a retina display mechanism such as iPad gives the worst matching performance indicating that it is not a good method to spoof a fingerphoto based biometric system.

- Table 4 shows the results of the performance of different matching algorithms captured in terms of their True accept rate at 1% FAR and 0.1% FAR. We see that the photo attack with iPad-Nokia has the least TAR for ScatNet+NN and ScatNet+RDF matching algorithms, and it is in accordance with Table 3 which provides the highest EER for iPad-Nokia.

4. How to Detect Spoofed Fingerphoto?

Now that it is experimentally established that it is possible to spoof a fingerphoto based system, in this section we study different features that can be used to detect spoofed images. A spoofing detection algorithm can be placed as a preprocessing module in a fingerphoto matcher's pipeline, which will discard spoofed images and process only non-spoofed original images.

4.1. Spoof Detection Algorithm

In this research, we evaluate a baseline spoof detection algorithm to detect and distinguish between spoofed fingerphoto images and original fingerphoto images, so that the recognition systems can address the challenges of spoofing. From Figure 2, it can be observed that most of the spoofed images exhibit Moire texture patterns [13] due to the property of the display devices. Such a texture is generally not observed if the capture mechanism directly captures the image of the finger. Hence, spoof detection is formulated as a binary classification problem using an SVM [7] to learn these texture patterns from a spoofed image. The texture patterns are extracted using LBP features [3, 8, 11]. To study the behavior of high definition display devices such as retina display, gradient based DSIFT [10] features and LUCID descriptor [17] are also independently used to learn an SVM. LUCID descriptors are recently found to provide successful performance in the domain of mobile biometric liveness detection [5]. Therefore, experiments with three anti-spoofing approaches present the baseline results on the proposed spoofed fingerphoto database.

4.2. Experimental Protocol

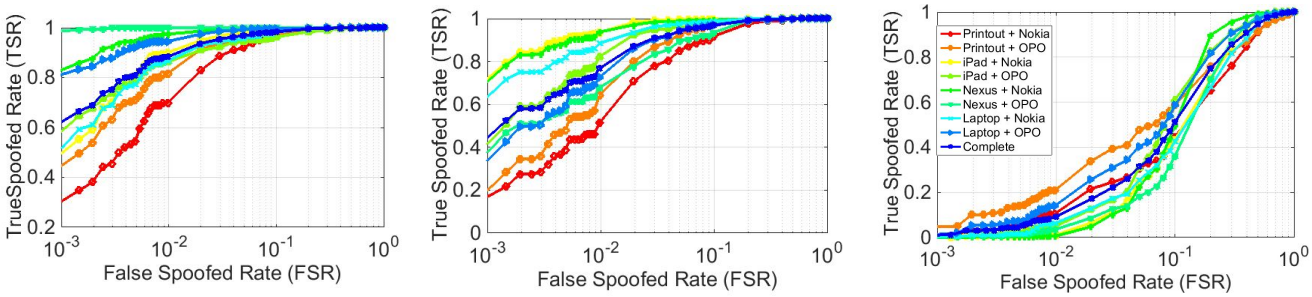
In this section, we experimentally evaluate the performance of baseline spoof detection algorithm using the proposed fingerphoto spoofed database. In the experiments, a 50% train-test data split is followed and a linear C-SVM (using lib-SVM [7] with $c = 1$) is used for training. The train data consists of (64 classes $\times$ 4 variations $\times$ $((2 \times 8)$ spoofed images + 6 original images that are not used for spoofing)). There are in-total 5632 images used for training with 4096 spoofed images and 1536 original images. The test data consists of images from (64 classes $\times 4$ variations $\times((2 \times 8)$ spoofed images + 8 original images including the ones used for spoofing)). Thus, the test data has 6144 images with 4096 spoofed images and 2048 original images.

4.3. Spoofing Detection Performance

The results using both LBP, DSIFT, and LUCID descriptors are presented in Table 5 and Table 6 and the ROC curves are shown in Figure 5. While using the complete test set, LBP + SVM gives the best spoofed fingerphoto detection performance with 3.71% EER when the complete

Spoof Attack	Display	Capture	True Accept Rate (%) @ FAR					
			ScatNet + L2		ScatNet + NN		ScatNet + RDF	
			0.1%	1%	0.1%	1%	0.1%	1%
Print Attack	Color Print	Nokia	53.46 ± 18.17	65.81 ± 13.83	97.03 ± 0.39	98.90 ± 0.58	91.65 ± 4.01	97.24 ± 2.10
		OPO	56.62 ± 20.10	65.16 ± 18.08	94.18 ± 2.95	98.32 ± 0.77	79.37 ± 11.52	96.63 ± 1.87
Photo Attack	iPad	Nokia	56.61 ± 27.24	67.53 ± 19.83	69.79 ± 7.47	85.60 ± 0.94	71.84 ± 6.98	91.17 ± 5.55
		OPO	50.55 ± 23.42	63.17 ± 12.63	78.68 ± 9.79	89.98 ± 5.83	93.87 ± 0.84	97.85 ± 1.06
	Nexus	Nokia	53.66 ± 21.62	70.92 ± 10.96	82.46 ± 1.05	91.87 ± 1.66	89.51 ± 2.42	96.37 ± 1.29
		OPO	53.26 ± 21.13	65.19 ± 15.04	79.00 ± 3.19	92.26 ± 1.25	84.13 ± 13.93	97.69 ± 1.41
	Laptop	Nokia	52.17 ± 15.59	65.20 ± 12.95	91.75 ± 4.31	97.66 ± 0.04	92.29 ± 2.40	99.08 ± 0.16
		OPO	53.62 ± 17.67	65.06 ± 11.68	93.64 ± 1.20	97.67 ± 0.58	96.72 ± 0.66	99.24 ± 0.15
Original images not used for spoofing			53.45 ± 25.80	63.18 ± 20.61	70.19 ± 12.03	91.04 ± 1.51	96.61 ± 3.16	99.66 ± 0.22
Original images used for spoofing			51.64 ± 26.51	60.31 ± 24.97	75.40 ± 5.69	89.56 ± 1.87	96.22 ± 1.90	99.56 ± 0.17

Table 4: Performance of different matching algorithms across the spoofed and non-spoofed subsets, captured in terms of TAR @ 1%FAR and 0.1% FAR



(a) LBP (b) DSIFT (c) LUCID

Figure 5: ROC curves showing the performance of LBP, DSIFT and LUCID with SVM to distinguish between spoofed and original images.

spoofed dataset is considered. Interestingly, it can be observed in Table 3 that using Nexus as the display mechanism yields very poor matching performance, while they provide the best anti-spoofing performance in Table 6. In general, those images that are not correctly matched with the matching algorithm are easily distinguished as spoof images by the SVM classifier, which is in accordance with the basic understanding of the problem. While the equal error rate is not very high, it is important that the spoofing detection errors are low at lower values of false accept rate as well. As shown in Table 6, existing descriptors commonly used in spoofing literature yield very poor results for fingerphoto detection. It is our assertion that this preliminary case study along with the availability of the large database provided in this research should encourage more researchers to work on this important problem.

Display	Capture	Equal Error Rate (%)		
		LBP + SVM	DSIFT + SVM	LUCID + SVM
Print	Nokia	6.05	9.17	26.95
	OPO	4.85	5.85	22.46
iPad	Nokia	3.12	1.90	23.91
	OPO	5.27	4.12	18.75
Nexus	Nokia	1.39	2.34	17.21
	OPO	0.24	8.00	22.85
Laptop	Nokia	4.48	3.29	25.17
	OPO	2.31	5.22	18.75
Complete		**3.71**	5.37	22.22

Table 5: EER (%) for LBP, DSIFT, and LUCID with SVM to distinguish between spoofed and original images.

		True Accept Rate (%) @ 0.1% FAR		
Display	Capture	LBP + SVM	DSIFT + SVM	LUCID + SVM
Print	Nokia	30.08	16.60	0.98
	OPO	44.34	19.33	4.88
iPad	Nokia	49.22	71.48	0.00
	OPO	58.01	40.82	0.20
Nexus	Nokia	82.81	70.11	0.00
	OPO	98.63	36.91	0.39
Laptop	Nokia	50.98	62.89	0.20
	OPO	81.05	33.00	1.37
Complete		**61.89**	43.89	1.07

Table 6: TAR@0.1% FAR for LBP, DSIFT, and LUCID with SVM to distinguish between spoofed and original images.

5. Conclusion and Future Work

With increasing usage of smartphones in daily lives, fingerphotos can be used as a viable approach for authentication; however, it is important to understand the implications of spoofing attempts on fingerphoto recognition. This research establishes the possibility of spoofing a smartphone camera based fingerphoto based biometric system. To be able to study and address different spoofing challenges, a new database is created containing $8,192$ images with respect to two different spoofing attacks: print and photo attack, and eight different spoofing mechanisms including iPad, mobile, laptop, and printouts. This database along with the experimental protocol will be made publicly available to promote research in this important problem. Further, we evaluated different features such as LBP, DSIFT, and LUCID combined with a learning algorithm to classify spoofed and original images. We observed that using LBP features yields as low as 3.7% EER on the database. As a future work, we plan to develop an improved anti-spoofing matching algorithm for fingerphotos.

References

[1] Chaos computer club breaks apple touchid. https://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid. Accessed: 2016-01-18.

[2] Laptop face-recognition tech easy to hack, warns black hat researcher. http://www.computerworld.com/article/2531298/windows-pcs/laptop-face-recognition-tech-easy-to\-hack--warns-black-hat-researcher.html. Accessed: 2016-01-18.

[3] T. Ahonen, O. Univ., A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. In *Transactions on Pattern Analysis and Machine Intelligence*, pages 2037–2041. IEEE, 2006.

[4] Z. Akhtar, G. Fumera, G. L. Marcialis, and F. Roli. Evaluation of serial and parallel multibiometric systems under spoofing attacks. In *BTAS*, pages 283–288. IEEE, 2012.

[5] Z. Akhtar, C. Micheloni, C. Piciarelli, and G. L. Foresti. Mobio_livdet: Mobile biometric liveness detection. In *AVSS*, pages 187–192. IEEE, 2014.

[6] L. Breiman. Random forests. In *Machine Learning, vol. 45, no. 1,*, pages 5–32, 2011.

[7] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. In *ACM TIST*, pages 1–27. IEEE, 2011.

[8] N. Erdogmus and S. Marcel. Spoofing in 2d face recognition with 3d masks and anti-spoofing with kinect. In *BTAS*, pages 1–6. IEEE, 2013.

[9] Z. W. Gang Pan, Lin Sun and S. Lao. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In *ICCV*, pages 1–8, 2007.

[10] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157. IEEE, 1999.

[11] J. Määttä, A. Hadid, and M. Pietikainen. Face spoofing detection from single images using micro-texture analysis. In *IJCB*, pages 1–7. IEEE/IAPR, 2011.

[12] D. J. Ohana, L. Phillips, and L. Chen. Preventing cell phone intrusion and theft using biometrics. In *Security and Privacy Workshops*, pages 173–180. IEEE, 2013.

[13] K. Patel, H. Han, A. Jain, and G. Ott. Live face video vs. spoof face video: Use of moire patterns to detect replay video attacks. In *ICB*, pages 98–105. IAPR, 2015.

[14] A. Sankaran, A. Malhotra, A. Mittal, M. Vatsa, and R. Singh. On smartphone camera based fingerphoto authentication. In *BTAS*, pages 1–7. IEEE, 2015.

[15] L. Sifre and S. Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *CVPR*, pages 1233–1240. IEEE, 2013.

[16] C. Stein, V. Bouatou, and C. Busch. Video-based fingerphoto recognition with anti-spoofing techniques with smartphone cameras. In *BIOSIG*, pages 1–12. IEEE, 2013.

[17] A. Ziegler, E. Christiansen, D. Kriegman, and S. J. Belongie. Locally uniform comparison image descriptor. In *NIPS*, pages 1–9, 2012.