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].

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 applications1. 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
attacks, such as print attack and photo attack, on fingerprints to establish the extent to which fingerprints 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 fingerprint 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 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 fingerprint detection is a comparatively newer paradigm for biometric authentication, fingerprint spoofing has not been well explored in the literature. As there is no publicly available spoofed fingerprint database, we created one using the existing IIITD FingerPhoto Database [14]. As shown in Figure 3, the setup used to create spoofed fingerprint 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 fingerprintphotos, containing a total of 4096 images. There are 128 classes × 4 variations × 8 instances in the original dataset. The spoofed

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

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

2http://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.

<table>
<thead>
<tr>
<th>Spoof/Non-spoof</th>
<th>Display</th>
<th>Capture</th>
<th># Images</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print attack</td>
<td>Color print</td>
<td>Nokia</td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>Photo attack</td>
<td>iPad</td>
<td>Nokia</td>
<td>1024</td>
<td>6144</td>
</tr>
<tr>
<td></td>
<td>Nexus</td>
<td>Nokia</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Laptop</td>
<td>Nokia</td>
<td>1024</td>
<td></td>
</tr>
<tr>
<td>Non-spoofed</td>
<td>4096</td>
<td>4096</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

The 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 × 4 variations × 2 instances × 2 capture mechanisms × 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-
Figure 4: Receiver operating characteristic (ROC) curves for different spoofing and non-spoofing subsets establishing the effectiveness of different spoof attacks.

<table>
<thead>
<tr>
<th>Spoof Attack</th>
<th>Display</th>
<th>Capture 1</th>
<th>Capture 2</th>
<th>Capture 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print Attack</td>
<td>Nokia</td>
<td>11.18 ± 0.35</td>
<td>1.02 ± 0.46</td>
<td>1.37 ± 0.66</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>12.55 ± 3.37</td>
<td>1.30 ± 0.43</td>
<td>1.63 ± 0.54</td>
</tr>
<tr>
<td>iPad</td>
<td>Nokia</td>
<td>7.66 ± 1.35</td>
<td>4.71 ± 0.48</td>
<td>2.53 ± 1.08</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>14.11 ± 1.88</td>
<td>3.65 ± 0.93</td>
<td>1.52 ± 0.66</td>
</tr>
<tr>
<td>Nexus</td>
<td>Nokia</td>
<td>10.41 ± 1.09</td>
<td>3.68 ± 0.86</td>
<td>1.94 ± 0.48</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>11.47 ± 0.46</td>
<td>3.31 ± 0.28</td>
<td>1.41 ± 0.38</td>
</tr>
<tr>
<td>Laptop</td>
<td>Nokia</td>
<td>13.54 ± 2.76</td>
<td>1.83 ± 0.12</td>
<td>0.96 ± 0.11</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>13.70 ± 2.15</td>
<td>1.50 ± 0.28</td>
<td>0.86 ± 0.10</td>
</tr>
<tr>
<td>Original images not used for spoofing</td>
<td>Nokia</td>
<td>11.46 ± 1.53</td>
<td>4.23 ± 0.34</td>
<td>0.48 ± 0.22</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>11.43 ± 1.77</td>
<td>3.58 ± 0.62</td>
<td>0.80 ± 0.15</td>
</tr>
</tbody>
</table>

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

representation 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 × 4 variations × 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 × 4 variations × 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 Scat-
Net 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 sub-
sets across the same algorithm suggests about the effective-
ness 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 ef-
  fective 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 indicat-
  ing that it is not a good method to spoof a fingerphoto
  based biometric system.
- Table 4 shows the results of the performance of dif-
  ferent 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 possi-
ble 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 finger-
photo 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 prop-
erty of the display devices. Such a texture is generally not
observed if the capture mechanism directly captures the im-
age 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 perfor-
mance of baseline spoof detection algorithm using the pro-
posed 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 × 4 variations × ((2 × 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 ×4 variations
×((2 × 8) spoofed images + 8 original images including the
ones used for spoofing)). Thus, the test data has 6144 im-
ages with 4096 spoofed images and 2048 original images.

4.3. Spoofing Detection Performance

The results using both LBP, DSIFT, and LUCID descrip-
tors 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 de-
tection performance with 3.71% EER when the complete
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

<table>
<thead>
<tr>
<th>Spoof Attack</th>
<th>Display</th>
<th>Capture</th>
<th>True Accept Rate (%) @ FAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print Attack</td>
<td>Color</td>
<td>Nokia</td>
<td>ScatNet + L2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1%</td>
<td>1%</td>
</tr>
<tr>
<td>Print</td>
<td>Print</td>
<td>OPO</td>
<td>53.46 ± 18.17</td>
</tr>
<tr>
<td>iPad</td>
<td>Nokia</td>
<td>OPO</td>
<td>56.62 ± 20.10</td>
</tr>
<tr>
<td>Nexus</td>
<td>OPO</td>
<td>50.55 ± 23.42</td>
<td>63.17 ± 12.63</td>
</tr>
<tr>
<td>OPO</td>
<td>53.66 ± 21.62</td>
<td>70.92 ± 10.96</td>
<td>82.46 ± 1.05</td>
</tr>
<tr>
<td>Original images not used for spoofing</td>
<td>52.17 ± 15.59</td>
<td>65.20 ± 12.95</td>
<td>91.75 ± 4.31</td>
</tr>
<tr>
<td>Original images used for spoofing</td>
<td>53.52 ± 17.67</td>
<td>65.06 ± 11.68</td>
<td>93.64 ± 1.20</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Display</th>
<th>Capture</th>
<th>Equal Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print</td>
<td>Nokia</td>
<td>6.05</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>4.85</td>
</tr>
<tr>
<td>iPad</td>
<td>Nokia</td>
<td>3.12</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>5.27</td>
</tr>
<tr>
<td>Nexus</td>
<td>Nokia</td>
<td>1.39</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>0.24</td>
</tr>
<tr>
<td>Laptop</td>
<td>Nokia</td>
<td>4.48</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>2.31</td>
</tr>
<tr>
<td>Complete</td>
<td></td>
<td>3.71</td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>Display</th>
<th>Capture</th>
<th>LBP + SVM</th>
<th>DSIFT + SVM</th>
<th>LUCID + SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print</td>
<td>Nokia</td>
<td>30.08</td>
<td>16.60</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>44.34</td>
<td>19.33</td>
<td>4.88</td>
</tr>
<tr>
<td>iPad</td>
<td>Nokia</td>
<td>49.22</td>
<td>71.48</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>58.01</td>
<td>40.82</td>
<td>0.20</td>
</tr>
<tr>
<td>Nexus</td>
<td>Nokia</td>
<td>82.81</td>
<td>70.11</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>98.63</td>
<td>36.91</td>
<td>0.39</td>
</tr>
<tr>
<td>Laptop</td>
<td>Nokia</td>
<td>50.98</td>
<td>62.89</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>OPO</td>
<td>81.05</td>
<td>33.00</td>
<td>1.37</td>
</tr>
<tr>
<td>Complete</td>
<td></td>
<td>61.89</td>
<td>43.89</td>
<td>1.07</td>
</tr>
</tbody>
</table>

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


