Abstract—Fingerprint recognition for automated border control and other high-security applications needs robust integrated anti-spoofing capability. Facing the threat of presentation attacks, two key challenges to be solved are sensor interoperability and robustness versus new fabrication materials. This paper proposes convolutional neural networks for this task and presents an exhaustive comparison on latest LivDet 2011 and 2013 databases. Apart from classical classification nets, also metric-based deep siamese networks are evaluated learning a distance metric enforcing live-spoof pairs to be of higher distance than live-live pairs. This is useful for attended enrollment scenarios where a live gallery image is available (e.g. trusted-source fingerprint pairs. This is useful for attended enrollment scenarios where a live gallery image is available (e.g. trusted-source fingerprint pairs. This is useful for attended enrollment scenarios where a live gallery image is available (e.g. trusted-source fingerprint pairs. This is useful for attended enrollment scenarios where a live gallery image is available (e.g. trusted-source fingerprint pairs.

I. INTRODUCTION

With their widespread use for biometric identification, fingerprint systems are a key target for presentation attacks. Presentation Attack Detection (PAD) modules classify biometric samples as either live (non-spoof) or fake (spoof), trying to detect attempts to interfere with biometric systems’ operation (ISO/IEC 30107-1). Due to covert acquisition and large variety of materials for presentation attack instruments (fake fingers) fingerprints are largely exposed. Since PAD algorithms are typically based on features specific for the fabrication material and sensor used for training, new fabrication materials and sensor diversity generally degrade spoof detection performance drastically [9]. Fabrication materials and live images likely appear different across sensors, affecting PAD cross-sensor performance and making robustness with regards to device diversity a highly challenging problem [8]. Recent works are looking at options to automatically adapt the PAD module to unseen new spoofing materials [12]. For robustness to new materials and sensor diversity, hand crafted features are not easy to generalize and prone to overfitting.

This paper investigates and compares deep convolutional neural networks (CNNs) [1] for PAD. Compared to previous work in this direction [4], networks with boosted accuracy are identified (CaffeNet [7] and GoogLeNet [14]), transfer-learning (fine-tuning using a pre-trained model) concepts to deal with limited amounts of training data are employed, and new siamese configurations using evidence about the template in question (an idea illustrated in [15]) are examined, see Figs.1.2. The paper presents an unseen evaluation focused on assessing generalization of methods: tests with unseen materials and sensor interoperability assessment.

The remainder is organised as follows: An introduction to related work is given in Section II. Tested CNNs are outlined in Section III. Exhaustive experiments conducted on multiple sensor-/material-specific sets of LivDet are presented in Section IV. Section V forms the conclusion.

II. RELATED WORK

Fake fingerprints can be made from a series of different materials, including silicone, latex, gelatin, play-doh, waxes, and wood glue, or even cadavers [8]. Presentation attack success depends on the type of sensing technology (e.g., optical vs. capacitative). In the effort to improve the state-of-the-art presentation attack detectors, several liveness detection competitions (LivDet 2009, 2011 2013 and most recently, 2015) have been conducted [6]. Generally, presentation attacks can be detected by either gathering further evidence of the liveness of the subject (e.g. sensing blood circulation, or fluids

Fig. 1: CaffeNet-based PAD Module mode of operation.

Fig. 2: Siamese PAD Module integrating gallery sample.
- perspiration patterns - secreted when touching surfaces) or by passive methods detecting the presence of known materials (e.g., material structure, lack of high-resolution detail). This work focuses on the latter software-based methods, for which several methods, including Fourier Transform (FT), Local Binary Patterns (LBP), or Histograms of Invariant Gradients (HIG), have been investigated. Good overviews of PAD methods in fingerprint recognition can be found in [13], [8].

Although deep learning approaches have been applied for fingerprint liveness detection [4], presented techniques were largely hybrid (combined with other classification techniques) and no pure CNNs have been evaluated for this task shedding light onto cross-sensor and robustness to new fabrication materials. Previously, Menotti et al. derived an efficient spoof detection system through deep representations [11]. They first learn a suitable CNN architecture, which is determined through a random search procedure involving hyperparameter optimization of the network. The candidate architecture is evaluated by executing linear SVM on the deep representation obtained by the considered net. Filter weights are optimized via back-propagation. Recently, Frassetto et al. [4] have examined fingerprint liveness detection using CNNs and LBPs, however they employ a hybrid approach feeding the net’s output into an SVM rather than exploiting the power of deep networks only with best reported accuracy of 95.2% using 50,000 samples for training using LivDet 2009, 2011 and 2013 datasets.

The approach in this paper is even further enhancing accuracy with much less available training samples (adhering to the strict LivDet 2013 protocol).

### III. CNNs for PAD

CNNs as deep instances of neural networks use multiple layers of non-linear processing units, each of which (single perceptron) estimate an output hypothesis $h_{w,b}(x) = f(w^T x)$ using output vector $x$ of the preceding layer. Within a training phase, using $k$ training samples $(x^i,y^i)$, $x^i \in \mathbb{R}^n, y^i \in \mathbb{R}^m$ a Matrix $W$ (and vector $b$) is learnt, such that $h_{W,b}(x) \approx y$ when applying forward propagation. Weights are learned through back-propagation, iteratively updating weights optimizing a cost function $J(W,b) = \frac{1}{m} \sum_{i=0}^{m} \frac{1}{2} \|y^i - W_{i,b}(x^i)\|^2 + \frac{1}{2} \delta(W)$, where $\delta(W)$ is the weight decay term. CNNs exploit stationarity in natural images using patch-wise convolution thereby drastically reducing the number of weights. We follow a fine-tuning strategy re-using pre-trained layers from these caffe\(^1\) reference models:

- **CaffeNet** [7]: This adapted Alexnet implementation follows the idea of establishing a large, deep CNN through 5 convolutional and 3 fully connected layers on 227x227 RGB image patches. The raw input image is transformed into gradually higher levels of representation (e.g., edges, local shapes, object parts). Higher levels of the hierarchy are formed by composition of lower level features, see Fig. 1. The layers of the CNN are organized such that the first two are subdivided into four sublayers each: convolution (conv), $\text{max}(x,0)$ rectified linear units (RELU), max pooling, and local response normalization.

- **GoogLeNet** [14]: This complex network of 22 layers (27 with pooling) for multi-scale Hebbian principle image processing is the best-performing net of the ImageNet\(^2\) Large-Scale Visual Recognition Challenge 2014, taking 224x224 RGB image patches. It employs spatially stacked inception modules with dimension reductions, approximating an optimal sparse structure by dense building blocks.

### A. Fine-tuning

Weights $W^{(i)}$ for different layers $i$ can be learned using different models, re-using weights as initial values for a different model. Thereby learning speed is increased and overfitting avoided, which is crucial for the limited amount of spoofing images. Presented networks were pre-trained using 1.2 million ImageNet natural images using the original 1000 classes output layer (which was replaced to binary classification, as in the Kaggle challenge\(^3\)) and stochastic gradient descent to optimize $J(W,b)$. As key adaptation we used a low learning rate ($10^{-4}$ to $10^{-6}$) running a stochastic gradient descent on the target loss function for 5000 iterations. Supervised pre-training and fine-tuning are especially effective when training data is scarce.

### B. Siamese Network

Siamese networks are neural networks where two branches processing image patches share exactly the same setup and weights [3], see also Fig. 2. The two branches’ outputs

\(^1\)http://caffe.berkeleyvision.org/

\(^2\)http://www.image-net.org/challenges/LSVRC/

\(^3\)https://www.kaggle.com/c/dogs-vs-cats
are connected (concatenated) and processed in yet another layer learning the similarity metric. Siamese networks are particularly interesting for PAD as they allow to learn a distance metric forcing low pairwise distance between live-live samples and higher distance between live-spoof samples. Input patterns are mapped into a target space by the function $G_w(X)$ parameterized by $w$, such that the similarity metric $E_w(X_1, X_2) = \|G_w(X_1) - G_w(X_2)\|$ is small if $X_1$ and $X_2$ are both live fingerprints, and large if $X_1$ is a live fingerprint and $X_2$ is a spoof fingerprint. $W$ is the shared parameter vector that is subject to learning (ground truth $Y = 0$ for live-live pairs $(X_1, X_2)$ and $Y = 1$ for live-spoof pairs $(X_1, X_2)$). In training all possible combinations live-live and live-spoof are considered. Instead of using final probability layers indicating class memberships, Siamese networks can be built by replacing these layers with a linear “feature” layer that produces a 2 dimensional vector “comparing” the two vectors. The learning process minimizes a discriminative loss function that drives the similarity metric to be small for pairs of live fingerprints (in both intra- and cross-sensor) and large for pairs live-spoof.

This approach reformulates the typical classification problem adopted for static fingerprint liveness detection which processes a single fingerprint image. Specifically, the proposed method assumes an attended enrollment scenario in which there is access to reference samples, for example the passport holder’s fingerprint stored on the chip of an ICAO 9303 travel document. Note, this reduces the joint PAD-and-recognition problem [2] from 8-classes (live/spoof template vs. live-spoof sample with same/different source) to 4 classes.

IV. EXPERIMENTAL STUDY

This study focuses on (1) comparing the performance of CNNs for PAD; (2) efficient parameter selection finding best setup for learning rate, fine-tuning layers, and iteration count in presence of a limited number of training samples; (3) robustness with unseen materials leaving out materials at the training stage; and (4) interoperability evaluating the impact of changes in sensors.

A. Dataset

Our experiments were conducted on the publicly available databases provided in the Fingerprint Liveness Detection Competition$^4$, LivDet 2009 [10], LivDet 2011 [16] and LivDet 2013 [6] keeping original Train and Test subdivisions, see Table I. Examples of fingerprint images and in particular material variability are illustrated in Fig. 3 and for sensor variability in Fig 4. These databases come with different sensors and spoofing materials, with a balanced 1:1 live/fake fingerprint ratio.

B. Evaluation Procedure

In order to evaluate robustness to new fabrication materials and device diversity, we closely followed the LivDet training and test set partition and employed latest ISO/IEC 30107 metrics for evaluating presentation attacks at sensor-level:

- **APCER:** Attack presentation classification error rate - incorrectly as normal classified attack presentations (false acceptance of spoof samples, ferrfake).
- **NPCER:** Normal presentation classification error rate - incorrectly as attack classified normal presentations (false rejection of live samples, ferrlive).

Rates refer to test set accuracy by choosing a threshold where APCER $\approx$ NPCER for the training set. Further the AUC (area under the curve) of the Receiver Operating Characteristic (ROC) is reported. LivDet trains each sensor separately, and by representing the spoof class with same set of fabrication materials during both training and testing.

For robustness to new fabrication materials, we exclude each of the fabrication materials from training which will be used only for testing, i.e. for latex tests training includes all the fabrication materials from training sets except latex, while test involves either all or just the left-out material (latex) for the particular sensor and database (Biometrika’13). For interoperability assessment a similar approach is taken, however fixing evaluations to one material (Gelatin). Training uses 3 and 4 sensors, testing involves multiple or individual sensors using the sets of LivDet11, LivDet13 as listed in Table I.

C. Parameter Selection Results

With regards to parameter estimation, we employed the separate Identix’09 database as validation set for the Siamese net (CaffeNet in siamese configuration). Table III lists experimental results for this net: few features (25) in the last layer before comparison are optimal and base learn rate (LR) should ideally be larger than for the classical CaffeNet (best result for LR $10^{-5}$), which makes sense given the much larger amount of available training images. CaffeNet and GoogLeNet

---

$^4$http://prag.diee.unica.it/
TABLE II: Robustness to unseen materials (All = Ecoflex, Gelatin, Latex, Modasil, WoodGlue) on Biometrika’13.

<table>
<thead>
<tr>
<th>Training</th>
<th>Testing</th>
<th>Siamese AUC</th>
<th>APCER</th>
<th>NPCER</th>
<th>CaffeNet AUC</th>
<th>APCER</th>
<th>NPCER</th>
<th>GoogLeNet AUC</th>
<th>APCER</th>
<th>NPCER</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>All</td>
<td>0.971 6.1%</td>
<td>7.8%</td>
<td></td>
<td>0.991 4.2%</td>
<td>2.9%</td>
<td></td>
<td>0.992 4.3%</td>
<td>2.5%</td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>Ecoflex</td>
<td>0.997 4.9%</td>
<td>7.8%</td>
<td></td>
<td>0.998 0.5%</td>
<td>2.9%</td>
<td></td>
<td>0.999 0%</td>
<td>2.5%</td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>Gelatin</td>
<td>0.905 19.9%</td>
<td>7.8%</td>
<td></td>
<td>0.978 9%</td>
<td>2.9%</td>
<td></td>
<td>0.981 10%</td>
<td>2.5%</td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>Latex</td>
<td>0.986 2.1%</td>
<td>7.8%</td>
<td></td>
<td>0.994 3.5%</td>
<td>3%</td>
<td></td>
<td>0.992 4.5%</td>
<td>2.5%</td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>Modasil</td>
<td>0.991 1.9%</td>
<td>7.8%</td>
<td></td>
<td>0.995 2.5%</td>
<td>3%</td>
<td></td>
<td>0.991 3%</td>
<td>2.5%</td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>WoodGlue</td>
<td>0.976 7.8%</td>
<td>5.9%</td>
<td></td>
<td>0.991 4.5%</td>
<td>2.9%</td>
<td></td>
<td>0.996 4%</td>
<td>2.5%</td>
<td></td>
</tr>
<tr>
<td>All without Ecoflex</td>
<td>All</td>
<td>0.982 6.1%</td>
<td>4.3%</td>
<td></td>
<td>0.999 1.1%</td>
<td>2.1%</td>
<td></td>
<td>0.993 3.2%</td>
<td>4.4%</td>
<td></td>
</tr>
<tr>
<td>All without Ecoflex</td>
<td>Ecoflex</td>
<td>0.996 2.5%</td>
<td>4.3%</td>
<td></td>
<td>0.999 0%</td>
<td>2.1%</td>
<td></td>
<td>0.999 0%</td>
<td>4.4%</td>
<td></td>
</tr>
<tr>
<td>All without Gelatin</td>
<td>All</td>
<td>0.986 4.7%</td>
<td>4.7%</td>
<td></td>
<td>0.998 4.6%</td>
<td>0.4%</td>
<td></td>
<td>0.991 7%</td>
<td>7.5%</td>
<td></td>
</tr>
<tr>
<td>All without Gelatin</td>
<td>Gelatin</td>
<td>0.947 18.3%</td>
<td>5%</td>
<td></td>
<td>0.993 18%</td>
<td>0.4%</td>
<td></td>
<td>0.948 21%</td>
<td>2.2%</td>
<td></td>
</tr>
<tr>
<td>All without Latex</td>
<td>All</td>
<td>0.938 17.7%</td>
<td>7.5%</td>
<td></td>
<td>0.997 1.3%</td>
<td>3.5%</td>
<td></td>
<td>0.995 5.5%</td>
<td>1.5%</td>
<td></td>
</tr>
<tr>
<td>All without Latex</td>
<td>Latex</td>
<td>0.972 11.5%</td>
<td>7.5%</td>
<td></td>
<td>0.997 2%</td>
<td>3.5%</td>
<td></td>
<td>0.994 8%</td>
<td>1.6%</td>
<td></td>
</tr>
<tr>
<td>All without Modasil</td>
<td>All</td>
<td>0.979 3.9%</td>
<td>2.8%</td>
<td></td>
<td>0.999 0%</td>
<td>4.7%</td>
<td></td>
<td>0.993 4.8%</td>
<td>2.6%</td>
<td></td>
</tr>
<tr>
<td>All without Modasil</td>
<td>Modasil</td>
<td>0.999 0%</td>
<td>2.8%</td>
<td></td>
<td>0.999 0%</td>
<td>4.7%</td>
<td></td>
<td>0.993 3.5%</td>
<td>2.7%</td>
<td></td>
</tr>
<tr>
<td>All without WoodGlue</td>
<td>All</td>
<td>0.960 10%</td>
<td>4.5%</td>
<td></td>
<td>0.997 1.3%</td>
<td>1.3%</td>
<td></td>
<td>0.995 3.4%</td>
<td>3.5%</td>
<td></td>
</tr>
<tr>
<td>All without WoodGlue</td>
<td>WoodGlue</td>
<td>0.941 14.2%</td>
<td>4.5%</td>
<td></td>
<td>0.997 1%</td>
<td>2.2%</td>
<td></td>
<td>0.998 1.5%</td>
<td>3.5%</td>
<td></td>
</tr>
</tbody>
</table>

D. Performance of CNNs for PAD

We tested material robustness on the Biometrika’13 database and with overall classification accuracy of 93.1% configurations were also tested on a separate validation set (Biometrika’11). For CaffeNet the best performance on the validation set was obtained at a base LR of $10^{-5}$ (and $10^{-4}$ for GoogLeNet, respectively) and both nets used 2 output features indicating spoof/live probabilities. Snapshots were taken after 5,000 iterations, however this parameter turned out to have less impact than an inappropriate base LR. Note, that obtained learning rates were lower than in original configuration due to starting from pretrained versions from ImageNet (see Sect. III-A).

E. Robustness to Unknown Materials

The test on unseen materials revealed, that AUC performance on testing all materials stayed the same or even slightly improved across different networks in 12/15 cases. However, tests on the left-out material performance stayed in the insignificant range of $-3.6\%$ to $+4.6\%$ of the original AUC values for all nets. It is worth mentioning, that obtained (Siamese), 96.5% (CaffeNet) and 96.6% (GoogLeNet) in the 2-class problem, all three tested networks achieved high performance in experimental tests, see Fig. 5. Table II lists experimental results in detail. For the baseline performance on different materials, it was interesting to see that there was a clear observable trend for a relative order of materials (Ecoflex, Modasil, Latex, WoodGlue, Gelatine) almost over the entire ROC. Interestingly, CaffeNet delivers very high EER performance while weakening a bit at the lower end of the ROC. With this regards GoogLeNet performed best and surprisingly even with relatively few images for finetuning (2,000 images for “All” instead of more than 2 million pairs for Siamese). This justified assumptions that nets adapt quite well to new representative samples limiting the additional benefit of pairwise-combinations.

TABLE III: Finding the best configuration for Siamese using validation set Identix’09.

<table>
<thead>
<tr>
<th>Nr</th>
<th>Net</th>
<th>Base LR</th>
<th>Features</th>
<th>Iterations</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Siamese</td>
<td>$10^{-7}$</td>
<td>50</td>
<td>7500</td>
<td>0.530</td>
</tr>
<tr>
<td>2</td>
<td>Siamese</td>
<td>$10^{-4}$</td>
<td>50</td>
<td>7500</td>
<td>0.947</td>
</tr>
<tr>
<td>3</td>
<td>Siamese</td>
<td>$10^{-4}$</td>
<td>50</td>
<td>5000</td>
<td>0.958</td>
</tr>
<tr>
<td>4</td>
<td>Siamese</td>
<td>$10^{-4}$</td>
<td>25</td>
<td>5000</td>
<td>0.970</td>
</tr>
</tbody>
</table>

TABLE IV: Interoperability Evaluation (4 S = Biometrika’13, Italdata’13, Sagem’11, Digital’11; 3 S = 4 S w/o Digital’11).

<table>
<thead>
<tr>
<th>Train Test</th>
<th>CaffeNet AUC</th>
<th>APCER</th>
<th>NPCER</th>
<th>GoogLeNet AUC</th>
<th>APCER</th>
<th>NPCER</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 S 3 Sensors</td>
<td>0.814 41%</td>
<td>8.5%</td>
<td></td>
<td>0.953 18%</td>
<td>7.3%</td>
<td></td>
</tr>
<tr>
<td>4 S 4 Sensors</td>
<td>0.809 55.9%</td>
<td>3.9%</td>
<td></td>
<td>0.911 25.8%</td>
<td>11%</td>
<td></td>
</tr>
<tr>
<td>3 S Biometrika’13</td>
<td>0.718 94.5%</td>
<td>5%</td>
<td></td>
<td>0.996 2.5%</td>
<td>3.5%</td>
<td></td>
</tr>
<tr>
<td>4 S Biometrika’13</td>
<td>0.610 96.5%</td>
<td>5.5%</td>
<td></td>
<td>0.991 2%</td>
<td>13.5%</td>
<td></td>
</tr>
<tr>
<td>4 S Italdata’13</td>
<td>0.972 4.5%</td>
<td>9.5%</td>
<td></td>
<td>0.932 8%</td>
<td>30.5%</td>
<td></td>
</tr>
<tr>
<td>4 S Sagem’11</td>
<td>0.910 54.5%</td>
<td>1.0%</td>
<td></td>
<td>0.922 41%</td>
<td>0.5%</td>
<td></td>
</tr>
<tr>
<td>4 S Digital’11</td>
<td>0.961 46.5%</td>
<td>0.5%</td>
<td></td>
<td>0.932 51.5%</td>
<td>0.5%</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4: Gelatin Spoof samples recorded with different optical sensors (from LivDet11 [16] and LivDet13 [6]).
1-APCER (True Spoof Accepts, %)  
NPCER (False Live Rejects, %)  
All  
Ecoflex  
Gelatine  
Latex  
Modasil  
W-Glue  

(a) Siamese  
(b) CaffeNet  
(c) GoogLeNet  

Fig. 6: Biometrika’13 ROCs for unseen materials (Training: All without specific material, Testing: specific material).

F. Sensor Interoperability

Finally, we tested sensor interoperability training networks with a single spoof-material for multiple sensors and testing on both the unseen sensor and all sensors, see Table IV. With AUCs as low as 0.814 for CaffeNet and 0.953 for GoogLeNet for the 3 sensors case and 0.809 and 0.911 for the 4 sensor case, respectively, it is evident that networks are less able to adapt to changes in sensors and raising this interoperability
issue. A further fact stressing the difficulty in this scenario is the limited number of training images that are generally available for the cross-sensor case (comp. Table I). Especially Biometrika 13 performance was much degraded in presence of other less challenging data sets and in particular the inability to detect attack presentations (high APCER) was evident for the trained CaffeNet network.

V. CONCLUSION AND FUTURE WORK

Among the 3 tested CNNs for PAD, GoogLeNet showed best performance learning fingerprint spoof materials with 4.3% APCER and 2.5% NPCER on Biometrika’13, closely followed by CaffeNet and Siamese - especially with regards to low NPCER. The successful ability to adapt to PAD problems using pre-training on ImageNet was shown, illustrating optimal parameters for the adaptation. All networks exhibit high robustness to unseen materials (-3.6% to +4.6% AUC deviation). Sensor interoperability had a more pronounced effect illustrating a slight advantage of CaffeNet over GoogLeNet. Compared to the other two nets, Siamese’ performance was inferior, yet it should be noted that for reasons of fairness the experiment did not consider live-live pairs of the same identity only, which could have boosted rates and is subject to future work. Further, it is planned to further customize CNNs with regards to processing speed.

ACKNOWLEDGMENT

The authors would like to thank Dr. Richard Souvenir and Junjie Shan for their support.

This work has been supported by the Modentity project and received funding from the Austrian Security Research Programme KIRAS (www.kiras.at), an initiative of the Federal Ministry for Transport, Innovation and Technology, Austria (www.bmvit.gv.at), under grant agreement no 845495.

REFERENCES