Fingerprint Presentation Attack Detection utilizing Time-Series, Color Fingerprint Captures

Richard Plesh
Clarkson University
Potsdam, NY, USA
pleshro@clarkson.edu

Keivan Bahmani
Clarkson University
Potsdam, NY, USA
bahmank@clarkson.edu

Ganghee Jang
Clarkson University
Potsdam, NY, USA
gjang@clarkson.edu

David Yambay
Clarkson University
Potsdam, NY, USA
yambayda@clarkson.edu

Ken Brownlee
Silk ID Systems Inc.
Santa Clara, CA, USA
ken@silkid.com

Timothy Swyka
Precise Biometrics
Potsdam, NY, USA
tim.swyka@precisebiometrics.com

Peter Johnson
Precise Biometrics
Potsdam, NY, USA
peter.johnson@precisebiometrics.com

Arun Ross
Michigan State University
East Lansing, MI, USA
rossarun@cse.msu.edu

Stephanie Schuckers
Clarkson University
Potsdam, NY, USA
sschucke@clarkson.edu

Abstract

Fingerprint capture systems can be fooled by widely accessible methods to spoof the system using fake fingers, known as presentation attacks. As biometric recognition systems become more extensively relied upon at international borders and in consumer electronics, presentation attacks are becoming an increasingly serious issue. A robust solution is needed that can handle the increased variability and complexity of spoofing techniques. This paper demonstrates the viability of utilizing a sensor with time-series and color-sensing capabilities to improve the robustness of a traditional fingerprint sensor and introduces a comprehensive fingerprint dataset with over 36,000 image sequences and a state-of-the-art set of spoofing techniques. The specific sensor used in this research captures a traditional gray-scale static capture and a time-series color capture simultaneously. Two different methods for Presentation Attack Detection (PAD) are used to assess the benefit of a color dynamic capture. The first algorithm utilizes Static-Temporal Feature Engineering on the fingerprint capture to generate a classification decision. The second generates its classification decision using features extracted by way of the Inception V3 CNN trained on ImageNet. Classification performance is evaluated using features extracted exclusively from the static capture, exclusively from the dynamic capture, and on a fusion of the two feature sets. With both PAD approaches we find that the fusion of the dynamic and static feature-set is shown to improve performance to a level not individually achievable.

1. Introduction

Biometric characteristics are valuable for recognizing individuals since they are distinctive between individuals and are inherent to every person. Since biometric characteristics can be publicly observed, systems cannot assume security from secretiveness. For example, latent fingerprints can be easily lifted from many different surfaces which can be used to create fake biometrics and attack the system, also called presentation attacks (PA). Therefore, unlike traditional password systems, the presentation of a matching biometric characteristic cannot be the only requirement for recognition. A second step is needed to verify that the biometric characteristics are being presented by an authorized, live, and present source, and this is called Presentation Attack Detection (PAD) [12].

The current landscape of PAD methods consists of hardware-based and software-based approaches. The former requires the collection of further finger information via additional sensor hardware, such as spectral absorption, skin temperature, and pulse oximetry [1]. Software-based approaches do not require additional hardware because they utilize the information contained in the digital image cap-

978-1-7281-3640-0/19/$31.00 ©2019 IEEE

tured by the sensor [12]. Software approaches are further divided into static and dynamic properties. Static properties come from a single fingerprint image, while dynamic are derived from a time-series capture. Researchers have developed many different methods of extracting features from either static or dynamic captures.

Researchers from the University of Utah and MIT studied the effect of pressure on the hemodynamic state of the fingertip. They found that the application of force produced blood perfusion coloration patterns beneath the fingernail that were common among all people in general [10]. Previous work by Wei-Yun Yau, Hoang-Thanh Tran, Eam-Khwang Teoh, and Jian-Gang Wang demonstrated the viability of detecting fake fingers using finger color change by developing and testing a method of measuring blood perfusion when a finger is applied to a sensor [16]. The data set used for testing was very small, was not generated using a commercial fingerprint sensor, and contained only one type of spoof finger. The research also does not evaluate any other dynamic changes that could be used for PAD. Our research addresses each of these shortcomings by developing and applying a collection of dynamic color features to a large and diverse data set generated using a sensor from SilkID Systems Inc.

Live fingerprints have unique properties in their fingerprint ridge signals due to the buildup of perspiration moisture. Previous publications have presented a low-cost software approach for time-series PAD that can utilize the perspiration phenomena in captures [11], [5]. Since perspiration is a relatively difficult psychological phenomenon to spoof, multiple methods have been developed for identifying perspiration for the purpose of spoof detection. In [15], perspiration is detected using a wavelet transform on a single image (static capture) fingerprint ridge signal. In [11], perspiration is detected using various measures of change in the ridge signal over a multi-frame fingerprint capture (dynamic capture).

The research could benefit from a more diverse data set however. The proposed algorithms also have practical limitations because time-series data rarely remain uniform enough to generate a meaningful difference image. This irregularity is due to the finger deforming and shifting during the capture period. In order to apply these techniques more successfully, we generate the feature value difference between frames, rather than extract features from a difference image, and test using a large data set.

An intensity-based approach has been applied by performing an image histogram of the fingerprint capture center. The "square area" of analysis is manually defined to avoid large scars and encompass an interesting part of the fingerprint [14]. This method has potential robustness advantages since the feature values should be resistant to skewing caused by smudged or shifting prints. In our research we utilize an automated and dynamic definition of the area of analysis, device-specific algorithm tuning, and a more diverse data-set for training. Additionally, time series features in this paper take into account shifting of the finger over the capture period by normalizing for each frame by ridge signal as well as foreground and full fingerprint ridge.

Local textural features have also been used for liveness detection [7]. These descriptors perform analysis on small patches of the fingerprint surface, in contrast to the global descriptors that use the entire image.

The analysis of skin distortion for spoof detection was performed in [2], [17], [3]. The technique in [2] requires a user to apply finger in a twisting motion with pressure while the sensor conducts a time-series capture. The research found that fake fingers are more rigid than human skin, with much less distortion. An approach that could measure distortion from a normal finger application (without twisting motion) would have practical ergonomic benefits for the user.

While previous work has explored dynamic features, this paper extends prior work by including advanced dynamic features based on color, evaluation on a large data set of over 36,000 image sequences of live fingerprints and a diverse set of spoof types, and feature-level fusion between dynamic and static features generated simultaneously.

This paper explores the utility of a color time-series fingerprint capture to fingerprint Presentation Attack Detection (PAD). We present the performance of state-of-the-art PAD algorithms based on a traditional gray-scale static (one frame) fingerprint capture, the state-of-the-art PAD algorithm for a color time-series (dynamic) capture, and a fusion of the two approaches. The dynamic PAD algorithms consist of novel features fused together with those from previous research while the static PAD algorithms are based on techniques introduced in prior papers. For training and testing of the algorithms we compose a diverse data set of live and spoof fingerprints using state-of-the-art spoofing techniques.

2. Dynamic and Color Fingerprint

While a static fingerprint capture gives us valuable liveness information, certain attributes of vitality cannot be captured effectively with a single gray-scale fingerprint capture. A dynamic color capture from a fingerprint sensor has the potential to provide access to difficult to spoof physiological signs of vitality, such as the displacement of blood in the finger when pressed. A time-series PAD method could also provide additional information on the spread of perspiration and the deformation of the finger over time. By utilizing the capability of the sensor to simultaneously capture a static and color time-series fingerprint (called hereon, fast frame rate or FFR), we explore the value of dynamic information to PAD performance.

State-of-the-art dynamic fingerprint capture PAD and baseline static fingerprint PAD are tested in isolation to evaluate the relevancy of the information in each capture to liveness detection. Last of all, the features extracted from each type of capture are fused together and evaluated to determine the novelty of the dynamic information to PAD. We hypothesize that a relatively high performance from a fusion method is indicative that each method is capturing a significant amount of unique information.

The sensor used was manufactured by Silk ID Systems Inc. and had specialized firmware loaded to perform the FFR functions. This sensor is a conventional optical fingerprint scanner utilizing dark-field, frustrated-total-internal-reflection as its imaging technique. The touch surface is a glass prism. A conventional CMOS Image Sensor of 2 megapixels resolution and standard color filter is used to capture the image. Four white LEDs illuminate the finger. When an IR finger detection circuit detects the presence of an object on the touch surface, a burst of 8 images are captured at approximately 8 -10 frames per second.

The images analyzed for the PAD are in their raw configuration to preserve the highest resolution. Some inherent distortion due to the optical configuration is present. This is later corrected by software to yield a geometrically-correct image for fingerprint matching. There are twice as many green pixels in the standard Bayer pattern of CMOS image sensors, yielding the strong green appearance in the raw images. Figure 1 presents three different captures from the SilkID sensor. Only two frames of the FFR capture are shown. Also, in this paper we refer to the foreground as the portion of the scanner image that contains the fingerprint and the background as the surrounding blank space.

3. Feature Extraction

This section describes the details of the static and dynamic feature extraction process. We employed several hand engineered feature extraction methods as well as Convolutional Neural Networks (CNNs) for feature extraction.

3.1. Optimum Frame Selection

The sensor captures a sequence of 8 frames to acquire the finger of the subject. Since the PAD approach presented in this paper calculates dynamic liveness character using the differences between only two frames in the time-series, a method is needed to select the best two frames of the captured sequence. One frame near the beginning, and one the end. We found that the first few frames could have inconsistent amounts of finger surface in the capture while the ending few frames had very little dynamic character since the finger was largely settled. In other words, selecting an earlier beginning frame would beget dynamic change at the expense of consistency. To solve this trade-off, the first frame is chosen as the earliest non-blank frame, while the second is at least 625msec from the first.

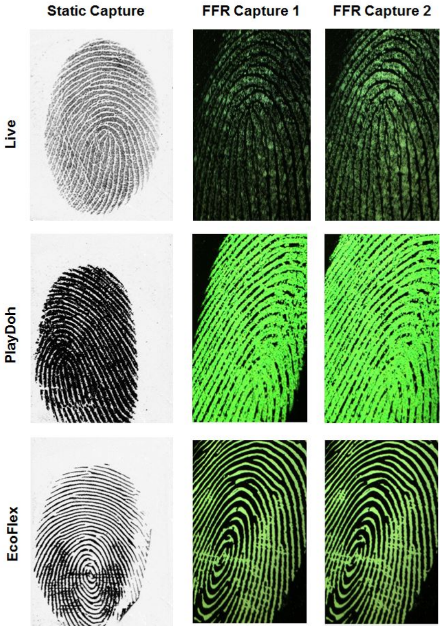

Figure 1. Static Image and selected frames from 3 different SilkID Captures from Live subject, PlayDoh PA, and EcoFlex PA

For the remainder of this paper, we use "1" to denote the beginning frame chosen by the algorithm, and "2" to denote the ending frame chosen by the algorithm. The time between frames is less than one second.

3.2. Dynamic Features

3.2.1 Dynamic Color features

When a live finger is pressed onto a sensor, a unique color change occurs, possibly due to the displacement of blood. The same mechanism makes our skin appear white when pressed up against a window. This dynamic character can be quantified and used for spoof detection. We have several different methods for extracting dynamic color information from the Fast Frame Rate (FFR) capture. In addition, three different portions of the fingerprint image are targeted for each extraction method and color ratio. The three portions are the fingerprint foreground, the fingerprint ridges, and the fingerprint ridge signal as shown in figure 2. The foreground pixels are calculated by way of a mask that blocks out areas of that do not contain the fingerprint. The identification of the ridge signal pixels and generation of the ridge signals is performed via the fingerprint enhancement routine described in previous work [15].

As the finger is applied to the sensor, we expect the values for each color channel to increase over time. In order to specifically target the change in finger color to white

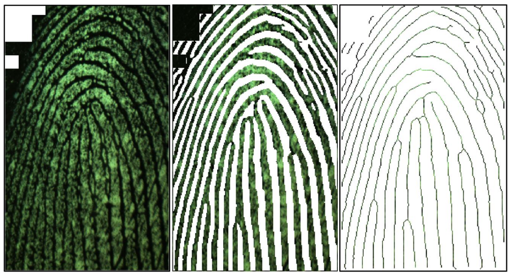

Figure 2. Fingerprint foreground, fingerprint ridge pixels, and the pixels used for generating the fingerprint ridge signal (left to right)

from the displacement of blood, we measure the change in green/red and blue/red ratio images between image 1 and image 2. The green/red and blue/red ratio images for the foreground pixels and ridge-signal are created via element-wise division between the images from each channel as shown in equation 1. Since the ridge signals may be shifted slightly over the course of the capture, a realignment routine uses correlation analysis to realign the signals. Also, the calculations on the ridge pixels are performed slightly different due to the difficulty in aligning the fingerprint from frame 1 to that of frame 2. Rather then using an element-wise division, the mean intensity is calculated on the raw color channels, and then divided to generate a color ratio measure as shown in equation 2. In both cases a 0.001 term is added to the denominator to prevent dividing by zero. The change in the "ratio-image" is calculated using difference in mean, ratio of means, and the square root of the sum of squared differences as shown in equation 3, 4, and 5 respectively. We expect the live subject to have a green/red or blue/red color ratio change greater in live over spoof. To explore other potential liveness phenomena experimentally, we perform dynamic change calculations on the green/blue ratio and each of the color channels individually as well. The outcomes of equations 3, 4, 5, and 6 are used as features for the classifier. Since there are multiple color channels and finger areas targeted, there are 12 features extracted from the fingerprint foreground, 12 features extracted from the ridge signal, and 7 features extracted from their ridge pixels.

$$
\begin{aligned}
FN_{CRI} &= FN_{clri} \oslash (FN_{clrj} + 0.001) && (1)\\
FN_{CRM} &= \frac{\overline{FN_{clri}}}{\overline{FN_{clrj}}} && (2)\\
diff &= \overline{F2_{ms}} - \overline{F1_{ms}} && (3)\\
ratio &= \frac{\overline{F2_{ms}}}{\overline{F1_{ms}}} && (4)\\
sumsquare &= \sqrt{\sum [F2_{ms} - F1_{ms}]^{2(ew)}} && (5)
\end{aligned}
$$

$$
Sequence_{euclid} = \sqrt{R_{diff}^2 + G_{diff}^2 + B_{diff}^2} \quad (6)
$$

where,

FN_{clr}: Image pixels from the Nth frame position (1: First Frame, 2: Last Frame) of channel clr (red, green, blue).
FN_{CRI}: Color Ratio Image for the Nth frame.
FN_{CRM}: Color Ratio Measure for the Nth frame. Needed for the ridge pixel features.
FN_{ms}: Measure from the Nth frame.
(ew): Signifies element-wise operation
C_{diff}: Difference in mean channel intensity between frames for channel color. Example: R_{diff}, G_{diff}, B_{diff}.
$diff$: Feature generated using the difference between two measures from each frame.
$ratio$: Feature generated using the ratio of measures from each frame.
$sumsquare$: Feature generated using the square root of the sum of the squared differences between the two frames.
$Sequence_{euclid}$: Feature generated from the Euclidean distance between the mean intensity values for the three channels.

3.2.2 Finger Shifting Features

In the data set we observed that spoofs generally have more shifting associated with them. We attribute this to the awkwardness of applying a fake finger. We quantify this change and use it as a feature for the classifier.

To detect the motion, we utilize the criteria outlined in equation 7. Lets call it delta image. The squared sum of pixel-wise difference captures the difference between images. The time difference between consecutive frames is 125 msec with marginal drift, so the difference between a neighboring pair can indicate the degree of movement of a finger. By limiting the squared difference to 255, the sum is more indicative of movement rather then color change.

$$
\begin{aligned}
press(x,y)_{frame(i)} &= (red(x,y)_{frame(i)} \\
&\quad - red(x,y)_{frame(i+1)})^2
\end{aligned}
$$

$$
Delta\ Image = \sqrt{\sum_{frame(i)} press(x,y)_{frame(i)}} \quad (7)
$$

3.2.3 Foreground/Background Mask Features

During our PAD processing we develop a mask to separate the foreground and background of the finger capture. We noticed that the mask itself contains information on liveness. It appears that spoof fingerprints generally have a smaller foreground and larger background as well as more movement in the foreground mask relative to spoofs. The foreground and background areas are generated via sum of the mask elements as shown in equation 8 and 9 respectively. The ratio of foreground to background in each frame, the ratio change of foreground/background, the difference to foreground/background ratio, and the shift in foreground mask are calculated from equations 10 to 13.

$$
\begin{aligned}
AN_{Fore} &= \sum(Mask) && (8)\\
AN_{Back} &= \sum(\sim\ Mask) && (9)\\
RN_{ForeBack} &= \frac{AN_{Fore}}{AN_{Back}} && (10)\\
RS &= \frac{R1_{ForeBack}}{R2_{ForeBack}+0.0001} && (11)\\
\Delta_{Back} &= A2_{Back} - A1_{Back} && (12)\\
MS &= \sum A1_{Fore} \oplus A2_{Fore} && (13)
\end{aligned}
$$

where,

$$Mask(n,m) = \begin{cases} 1 & \text{if } frame(n,m) \text{ is fingerprint} \\ 0 & \text{if } frame(n,m) \text{ is empty background} \end{cases}$$

AN_{Fore}: The Area of the Foreground mask.
AN_{Back}: The Area of the Background mask.
$RN_{ForeBack}$: The Ratio of Foreground to Background for Frame N
RS: Ratio of the Foreground/Background measure for Frame 1 to that of Frame 2
Δ_{Back}: The difference between the background area of Frame 1 and 2
MS: A measure of shifting between Frame 1 and 2 using XOR to detect differences in Mask Profile

3.2.4 Dynamic Fingerprint Foreground Intensity

Dynamic foreground features are designed to capture the global intensity changes in the dynamic capture of the fingerprint and are a modification of the features presented in Tan, et, al [14]. Since these features do not concern the finger color information specifically, the RGB dynamic capture is converted to grayscale for analysis. The previous paper manually selected a section in the center of the fingerprint capture for analysis. Since a practical implementation of this algorithm would require an automated selection of capture area, we dynamically determine the foreground of the capture for frequency analysis and normalize accordingly. Foreground selection allows us to observe the intensity information for the entire finger area, rather then just a small section, and improves robustness by diminishing capture variability. An image intensity histogram is then generated from the two selected frames. Static measures are extracted from a single image histogram while dynamic features are generated from the difference between the two histograms. The measures extracted from the histograms needed to be tuned to the specific SilkID sensor characteristics, such as the inversion of black and white. Through this method, 6 features are extracted from the classifier.

3.2.5 Dynamic Ridge Signal Perspiration Detection

When a live finger is in contact with a sensor, a dynamic perspiration pattern can be detected in the ridge signal. To extract the temporal changes in the ridge signal due to perspiration we rely on the method described in Parthasaradhi et al[11]. Just as in this paper, we extract 1 static and 6 dynamic features from the fingerprint ridge signal. These features use the Fourier transform of the ridge signal, total swing in ridge signal, growth ratio of signal peaks, growth of signal mean, percent change in signal variance, dry saturation change, and wet saturation change.

3.3. Static Features

3.3.1 Engineered Feature Extraction

Our baseline engineered feature algorithm extracts features from only the static image capture. These features include local binary pattern, fingerprint foreground intensity, and multi-resolution fingerprint ridge and valley texture analysis features from previous research. Local binary pattern (LBP) analysis has been commonly used for biometric recognition and spoof detection [7]. In this research LBP was performed for two radii (1 and 2 pixels). Fingerprint foreground intensity utilizes the relative frequencies of different intensity levels within the fingerprint capture. Multi-resolution fingerprint ridge and valley texture analysis uses the wavelet transform to decompose the ridge and valley signal into multi-scales. Features are then extracted from each scale to quantify the perspiration pattern. 72 features are generated from LBP analysis, 64 features from fingerprint foreground intensity, 14 features from multiresolution ridge analysis, and 14 features from multiresolution valley analysis.

3.3.2 Feature Extraction Using Convolutional Neural Networks

In this work, we employed Inception-V3 CNN model [13] trained on Imagenet dataset [6] to extract features from the static images. The model is implemented using Keras platform [4], where each static images is re-sized to 299 by 299

Table 1. Summary of collected samples from presentation attack using finger spoofs.

Type of PA attack		Captures
Mold	Material	
3D	Dragonskin	258
3D	Ecoflex	267
3D	Gelatin	201
3D	ModelMagic	258
3D	PlayDoh	203
3D	SillyPutty	265
3D	Wood glue	217
Dental	Ecoflex	3924
Dental	Gelatin	1543
Dental	ModelMagic	1484
Dental	PlayDoh	3190
Dental	SillyPutty	1840
Dental	Wood glue	2378
Dental	Latex BodyPaint	1245
N/A	Paper (2D print)	2893
N/A	Transparent film	1534
N/A	Live Finger	14892
Sum All		36592

pixels and 2048 bottleneck features are extracted using the CNN model.

3.4. Data Set

One of the distinct characteristics of our data set is the variety of fake fingers, both in terms of mold type and the materials which form the fake fingers. In terms of molds, there are 3 distinct categories: 3D molds from 3D printer, dental molds, and 2D printing.

The 3D molds are created through the use of a 2D fingerprint image. The 2D fingerprint image is binarized and converted into a surface using the binarized grayscale values. For the molds used in this dataset, the ridge values are extruded down and the valley values are kept at a depth of 0. This new surface is re-meshed using meshmixer to ensure the integrity of the mesh and re-sized based on the original dpi of the input image to ensure proper sizing. Each fingerprint is then embedded into a base block using Blender. This new fingerprint block is then printed using a Formlabs 1+ STL printer for high resolution and overall quality. After printing, the molds are cured in a homemade UV curing unit before use.

In case of dental molds, we prepared molds using Kerr extrude (Polyinylsiloxane) impression material during the live data collection. Human live fingers were pressed into the material for approximately 4 minutes. This high quality material is used for dental applications, has very short cure time, and does not shrink.

3D molds and dental molds have the common process when making fake fingers. Once molds are prepared, we applied materials as in Table 1 on top of the molds. The quality depends on the resolution of the printer, and captured image. Dental molds generally carry better quality over 3D molds because there are no loss from data conversion. In order to increase diversity of data, the samples from a single material may differ in hardness, transparency, and color.

The last category of fake fingers is from 2D prints. 2D printing does not use molds at all. Rather, a printer applies black ink onto white paper using a drum. The source of the image is from previous fingerprint collections from traditional fingerprint scanners. We used two types of material, paper and transparent film, from multiple printers.

We created molds from the fingerprint from right hand fingers in order to avoid taking too much time from participants. Each presentation attack instrument was captured two times.

In total, the dataset contains 14892 live fingerprints captures and 21700 spoof fingerprint captures from 450 subjects. 33 of the subjects are unique to live captures, 208 are unique to spoof captures, and 209 of the subjects are in both sets. This dataset will be made available by request for comparison purposes under an IRB-approved database release agreement.

3.4.1 Data Cleaning

In practice, we observe a significant amount of variation in subject finger application when no capture quality feedback mechanism or guide is used during capture. Improper use of the sensor can occasionally create captures with blank frames. While a feedback mechanism and guide has been developed and implemented in a subsequent collection for this sensor, the data was not ready in time for use in this paper. To clean the data of captures with blank frames, we developed an algorithm that identifies individual blank frames using the standard deviation of the middle row of pixels. Sequences that don't contain seven sequential non-blank frames are removed from our dataset. Table 1 shows the data-set post cleaning.

3.5. Classifier

We evaluated the classification performance of both static and dynamic features using a fully connected Deep Neural Network (DNN) assessed using a 10-fold cross validation process.

Two different DNNs namely *DNN1* and *DNN2* are respectively evaluated on hand engineered features and features extracted using the Inception model. No preprocessing is applied to training and the features are normalized to have zero mean and unit variance. Tables 2 and 3 present the architecture of both networks. Both networks are initialized using Xavier uniform initialization [8] and trained using the Adam optimizer [9] with cross entropy loss function. *DNN1* employs an adaptive learning rate starting at 0.001 with patience of 10 epochs and reduce factor of 0.1 trained for 50 epochs, While *DNN2* uses a fixed learning rate of 0.0005 and is trained for 60 epochs.

Layers	Size in	Size Out	Activation
FC1	216×1	400×1	Relu
FC2	400×1	400×1	Relu
FC3	400×1	2×1	Softmax

Table 2. Architecture of the DNN1

Layers	Size in	Size Out	Activation
FC1	2048×1	500×1	Relu
FC2	500×1	500×1	Relu
FC3	500×1	2×1	Softmax

Table 3. Architecture of the DNN2

4. Results

In Figure 3 we compare the performance of static, dynamic, and fusion approaches to PAD detection using multiple ROC curves. The top curve demonstrates the performance received when deploying a classifier on top of a fusion of static and dynamic engineered features. Next down characterizes classifier performance when using solely dynamic color capture features. Further down characterizes performance from a static engineered-feature classifier. At roughly the same performance level as the former, exists our plot of classifier performance on bottleneck ImageNet-trained CNN features. The two unique methods of static feature extraction yield such similar results for this dataset. We also observe that the fusion approach yields better performance then either static or dynamic methods alone, suggesting the use of static and dynamic approaches simultaneously yields synergistic benefits. In table 4 we provide the mean Attack Presentation Classification Error Rate (APCER) at 0.2% and 1.0% Bona Fide Presentation Classification Error Rate (BPCER) operating points as well as the standard deviation (STD) at 1.0% BPCER.

5. Discussion and Conclusion

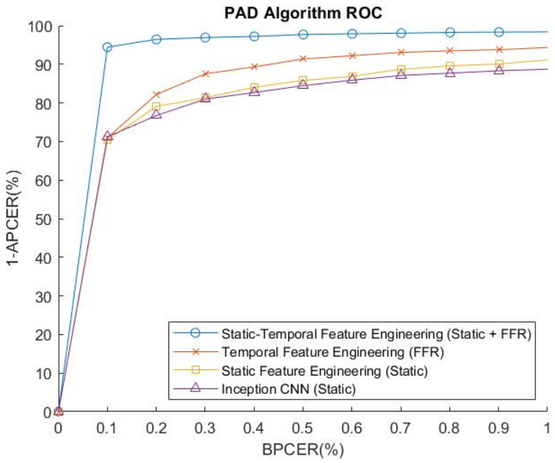

Figure 3. PAD algorithm ROC curve

PAD Algorithm	Mean APCER @0.2% BPCER	Mean APCER @1.0% BPCER	STD @1.0% BPCER
Static-Temporal Engineered Features (Static + FFR)	3.55%	0.626%	1.96%
Temporal Feature Engineering (FFR)	17.8%	1.10%	3.42%
Static Engineered Feature	20.9%	1.78%	7.82%
Static Features (Inception CNN)	23.3%	3.05%	4.05%

Table 4. PAD algorithm performance at 0.2% BPCER and 1.0% BPCER operating points

A novel approach to fingerprint presentation attack detection using a fusion of static and color time-series fingerprint capture is detailed in this paper. When tested with a large and diverse dataset, this approach has been shown to generate classification performance greater than that of the static classifier exclusively, on a challenging dataset of diverse spoof types. Hopefully this will encourage manufacturers to bring more sensors with this technology to the marketplace and push the capabilities of spoof detection to new heights in the coming future.

The presentation attack solutions presented in this paper were designed to be generally applicable to a sensor with similar capability. Due to the lack of capability in similar sensors we could not test the algorithm's generalization to different sensors directly.

6. Acknowledgements

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. 2017 - 17020200004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[1] K. Adair Nixon, V. Aimale, and R. Rowe. Spoof Detection Schemes. pages 403–423. Oct. 2007.

[2] A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni. Fake finger detection by skin distortion analysis. *IEEE Transactions on Information Forensics and Security*, 1(3):360–373, Sept. 2006.

[3] R. Cappelli, D. Maio, and D. Maltoni. Modelling Plastic Distortion in Fingerprint Images. In S. Singh, N. Murshed, and W. Kropatsch, editors, *Advances in Pattern Recognition ICAPR 2001*, Lecture Notes in Computer Science, pages 371–378. Springer Berlin Heidelberg, 2001.

[4] F. Chollet et al. Keras, 2015.

[5] B. DeCann, B. Tan, and S. Schuckers. A novel region based liveness detection approach for fingerprint scanners. In M. Tistarelli and M. S. Nixon, editors, *Advances in Biometrics*, Lecture Notes in Computer Science, pages 627–636. Springer Berlin Heidelberg, 2009.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, CVPR 2009*, pages 248–255. Ieee, 2009.

[7] L. Ghiani. *Textural Features for Fingerprint Liveness Detection*. PhD thesis, University of Cagliari, 2015.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[10] S. Mascaro and H. H. Asada. The Common Patterns of Blood Perfusion in the Fingernail Bed Subject to Fingertip Touch Force and Finger Posture. July 2006.

[11] S. T. V. Parthasaradhi, R. Derakhshani, L. A. Hornak, and S. A. C. Schuckers. Time-series detection of perspiration as a liveness test in fingerprint devices. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(3):335–343, Aug. 2005.

[12] S. Schuckers. Presentations and attacks, and spoofs, oh my. *Image and Vision Computing*, 55:26–30, Nov. 2016.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[14] B. Tan and S. Schuckers. *Liveness detection using an intensity based approach in fingerprint scanners*. 2005.

[15] B. Tan and S. Schuckers. Liveness Detection for Fingerprint Scanners Based on the Statistics of Wavelet Signal Processing. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 26–26, June 2006.

[16] W.-Y. Yau, H.-T. Tran, E.-K. Teoh, and J.-G. Wang. Fake Finger Detection by Finger Color Change Analysis. In S.-W. Lee and S. Z. Li, editors, *Advances in Biometrics*, Lecture Notes in Computer Science, pages 888–896. Springer Berlin Heidelberg, 2007.

[17] Y. Zhang, J. Tian, X. Chen, X. Yang, and P. Shi. Fake Finger Detection Based on Thin-Plate Spline Distortion Model. In S.-W. Lee and S. Z. Li, editors, *Advances in Biometrics*, Lecture Notes in Computer Science, pages 742–749. Springer Berlin Heidelberg, 2007.