International Journal of Advanced Trends in Engineering and Technology (IJATET)
Impact Factor: 5.665, ISSN (Online): 2456 - 4664
(www.dvpublication.com) Volume 2, Issue 1, 2017

FAKE IRIS - BIOMETRIC IDENTIFICATION BY FUTURISTIC COMPONENT BASED FAKE DETECTION USING UNUSAL APPROACH OVER SUPPLY VECTOR MACHINE

M. Sethupathy*, L. Rahunathan & M. Arumugam****
* Final Year Student, Department of Computer Applications, Kongu Engineering College, Perundurai, Erode, Tamilnadu
** Assistant Professor, Department of Computer Applications, Kongu Engineering College, Perundurai, Erode, Tamilnadu

Cite This Article: M. Sethupathy, L. Rahunathan & M. Arumugam, "Fake IRIS - Biometric Identification by Futuristic Component Based Fake Detection Using Unusal Approach Over Supply Vector Machine", International Journal of Advanced Trends in Engineering and Technology, Page Number 93-100, Volume 2, Issue 1, 2017.

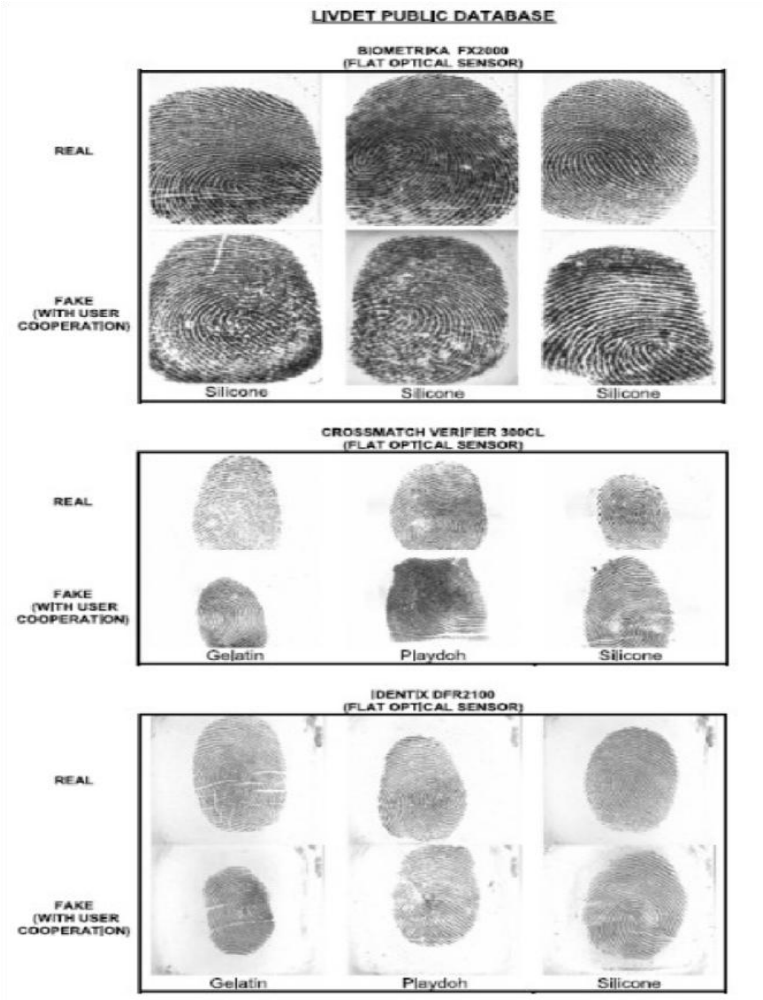
Abstract:
To ensure the actual occurrence of a real reasonable feature in contrast to a fake self manufactured synthetic or reconstructed sample is a significant problem in biometric confirmation, which requires the development of new and efficient protection procedures. In this paper, we present a novel software-based fake detection method that can be used in multiple biometric systems to detect different types of deceitful access attempts. The objective of the proposed system is to enrich the security of biometric appreciation frameworks, by adding live ness assessment in a fast, user-friendly, and non-intrusive manner, through the use of image worth assessment. The proposed approach presents a very low degree of difficulty, which makes it suitable for real-time applications, using 25 general image quality structures extracted from one image (i.e., the same acquired for authentication purposes) to distinguish between appropriate and impostor samples. The investigational results, obtained on publicly available data sets of fingerprint, iris, and 2D face, show that the proposed method is highly economical compared with other state-of-the-art attitudes and that the analysis of the common image quality of real biometric samples reveals highly valuable evidence that may be very efficiently used to distinguish them from fake traits.

Key Words: Weber Local Descriptor (WLD), Convolutional Neural Networks (CNN) & Local Phase Quantization (LPQ)

1. Introduction:

The basic aim of biometrics is to automatically discriminate subjects in a reliable manner for a target application based on one or more signals derived from physical or behavioral traits, such as fingerprint, face, iris, voice, palm, or handwritten signature. Biometric technology presents several advantages over classical security methods based on either some information (PIN, Password, etc.) or physical devices (key, card, etc.) [2].However, providing to the sensor a fake physical biometric can be an easy way to overtake the systems security. Fingerprints, in actual, can be easily deceived from common materials, such as gelatin, silicone, and wood glue [2]. Therefore, a safe fingerprint system must correctly distinguish a spoof from an authentic finger (Figure 1). Different fingerprint live ness detection algorithms have been proposed [3], [4], [5], and they can be broadly divided into two approaches: hardware and software. In the hardware approach, a exact device is added to the sensor in order to sense particular assets of a living trait such as blood pressure [6], skin distortion [7], or odor [8]. In the software approach, which is used in this study, fake traits are detected once the sample has been acquired with a standard sensor. The features used to distinguish between real and fake fingers are extracted from the image of the fingerprint. There are techniques such as those in [2] and [9], in which the features used in the classifier are based on specific finger print measurements, such as elevation strength, continuity, and clarity. In contrast, some works use common feature extractors such as Weber Local Descriptor (WLD) [10], which is a texture descriptor composed of differential excitation and orientation components. A new local descriptor that uses local fullness contrast (spatial domain) and phase (frequency domain) to form a bi dimensional contrast-phase histogram was proposed in [11]. In [12] two general feature extractors are compared: Convolutional Neural Networks (CNN) with random (i.e., not learned) weights (also explored in [13]), and Local Binary Patterns (LBP), whose multi-scale variant reported in [14] achieves good results in fingerprint live ness detection benchmarks. In contrast to more sophisticated techniques that use texture descriptors as features vectors, such as Local Phase Quantization (LPQ) [15], LBP with wavelets [16], and BSIF [17], their LBP implementation uses the original and uniform LBP coding schemes. Moreover, a variety of optional preprocesses techniques such as contrast normalization, frequency filtering, and region of interest (ROI) extraction were attempted without success. Augmented datasets [18] [19] are successfully used to increase the classifiers robustness against small variations by creating additional samples from image translations and horizontal reflections. In this study we extend the work presented in [12] by using a similar model from the well-known Alex Net [19], pre-trained on theILSVRC-2012 dataset[20], which contains over 1.2 million images and 1000 classes, and then fine-tuned on fingerprint images. We show that although the pre-trained model was designed to detect objects in natural images, fine-tuning it to the task of fingerprint live ness detection yields

better results than if trained the model using randomly initialized weights. Furthermore, we train our system using a larger pre-train model [21], VGG, the second place in the ILSVRC-2014 [20], to increase the accuracy of the classifier by another 2% in absolute values. Thus, the contributions of this study are three-fold: Deep networks designed and trained for the task of object recognition can be used to achieve state-of-the-art accuracy in fingerprint live ness detection. No specific hand engineered technique for the task of fingerprint live ness detection was used. Thus, we provide another success case of transfer learning for deep learning techniques. Pre-trained Deep networks require less labeled data to achieve good accuracy in a new task. Dataset augmentation helps to increase accuracy not only for deep architectures but also for shallow techniques such as LBP.



2. Methodology:

Relocation Learning is a enquiry problem in machine culture that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. Fig. 2. Some images from the Image NET dataset used to pre-train then networks. Despite their difference to the fingerprint images, pre-training with natural images do help in the task of fingerprint live ness detection. Fig. 3. Illustration of the models used in this study. The boxes in red are theonly layers that are different from the original VGG-19 and Alex net models. In this study, we showed that it is possible to achieve state of-the-art fingerprint live ness detection by using models that were originally designed and trained to detect objects in natural images (such as animals, car, people). The same idea is explored in [22], for which the authors achieved state of the art performance in CIFAR-10, Flicker Style Wiki paintings benchmarks using a pre-trained convolutional network. One important difference from their experiments to ours is that all the datasets they used contain similar images to the Image NET dataset (Figure 2), such as objects and scenes. In our study, fingerprint images were used, which differ significantly from those of other domains.

A. Models: Table I describes the models in this study. All of them used dataset augmentation. Additionally, we show the architecture of the models in Figure 3. For CNNVGG and CNN-Alex net, the architecture is the same as described in [20] and [19] respectively, except that we replaced the last 1000-unit soft max layer by a 2unit soft max layer (shown in red in the figure), so the network can output the 2 classes (if the image is real or fake)

instead of the original 1000 classes that the networks were designed for. For the CNN-Random the architecture is different for each dataset and it was chosen via an extensive grid-search as described in [12].

B. Convolutional Networks: Convolutional Networks [23] have demonstrated state-of the-art performance in a variety of image recognition bench Model Name Pipeline Description CNN- VGG 16 Convolutional Layers + 3 Fully Connected Layers Pretrained model from [20] and fine tuned using live ness detection datasets. CNN-Alex net 8 ConvolutionalLayers + 3 Fully Connected Layers Pre-trained model from [18] and fine tuned using live ness detection data sets. CNN-Random CNN-Random +PCA + SVM Features are extracted using Convolutional Networks. The feature vector is reduced using PCA and then fed into a SVM classifier using (Gaussian) RBF kernel. LBP LBP + PCA + SVM Features are extracted using LBP. The feature vector is reduced using PCA and then fed into a SVM classifier with (Gaussian) RBF kernel.

Summary of the Models Used in this Study: Marks, such as MNIST [24], CIFAR-10 [24], CIFAR100 [24], SVHN [24], and Image Net [25]. A classical convolutional network is composed of alternating layers of convolution and local pooling (i.e., subsampling). The aim of a convolution allayer is to extract patterns initiate within resident regions of the in putted similes that are common throughout the dataset by convolving a template over the inputted image pixels and out putting this as a feature map c, for each filter in the layer. A non-linear task f(c) is formerly pragmatic element-wise to each feature map c: a = f(c). A range of functions can be used for f(c), with max(0; c) a common choice. The result in activations f(c) are then passed to the pooling layer. This aggregates the information within a set of small local regions, R, producing a pooled feature map s (normally of smaller size)as the output. Denoting the aggregation function as pool (), for each feature map c we have: sj= pool(f(ci))8i 2 Rj, where $R_j$is the pooling region j in feature map c and i is the catalog of every division within it. Among the countless types of amalgamating, max-pooling is generally used, which selects the maximum value of the region R_j.The drive behind pooling is that the activations in the pooled map s are less searching to the precise locations of structures within the image than the original feature map c. In a multi-layer model, the convolutional layers, which take the collective maps as input, can thus extract features that are increasingly invariant to local revolutions of the input image [26] [27]. This is important for ordering tasks, since these transformations obfuscate the object identity. Achieving invariance to changes in position or lighting conditions, robustness to clutter, and compactness of representation, are all common goals of pooling.

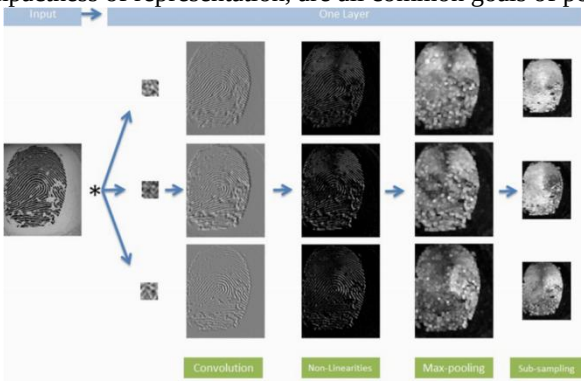


Figure 4 illustrates the feed-forward pass of a single layer convolutional network. The input sample image will convoluted with three random filters of size 5x5 (enlarged to make visualization easier), producing 3 elaborate images, which are then subject to nonlinear function max(x; 0), followed by a max-pooling operation, and subsampled by a factor of 2. In this study we compared three different models of convolutional networks. The first one, CNN-Random, uses only random filter weights draw from a Gaussian distribution. Although the filter weights can be learned, filters with random weights can perform well and they have the advantage that they do not need to be learned [28] [29] [30] . The architecture of the model is the same as that used in [12]. It uses convolutional network with random weights as the feature extractor, the dimensions are further reduced using PCA and a SVM classifier with RBF kernels used as the classifier. An extensive search for hyper parameter fine-tune was performed automatically on more than 2000 combinations of hyper-parameters The best hyper-parameters were chosen per sensor and per dataset (ex. Biometrika 2009) Bimetrika 2011 through validation method [31] which used the training dataset of each sensor in each Liv Det dataset (2009, 2011, 2013).

Pipeline
Element
Hyper-parameter Range
CNN-Random # Layers 1, 2, 3, 4, 5
CNN-Random # Filters (in each layer) 32, 64, ..., 2048
CNN-Random Filter Size Convolution 5x5, 7x7, ..., 15x15
CNN-Random Filter Size Pooling 3x3, 5x5, 7x7, 9x9
CNN-Random Stride (reduction factor)
2, 3, ..., 7

LBP Coding Standard or Uniform
LBP # Images Divisions 1x1 (no division), 3x3,
5x5, 7x7
PCA # Components 30, 100, 300, 500, 800,
1000, 1300
SVM Regularization Parameter
C
0.1, 1, ..., 105
SVM Kernel coefficient 107, 106, ..., 101 TABLE II

Range of Hyper-Parameters Searched for the CNN- Random and Lbp Pipelines: The second model, CNN-Alex net, is very similar to Alex Net [19], pre-trained on the ILSVRC-2012 dataset. This model won both classification and localization tasks in the ILSVRC-2012 competition. Their trained model has been used to improve accuracy in a variety of other benchmarks such as CIFAR-10, CIFAR-100. The pre-trained network provides a good starting point for learning the network weights for other tasks, such as fingerprint live ness detection. The third model, CNN-VGG, is very similar to the one used in [21], a 19 layer CNN which achieved the second place in the detection task of the Image Net 2014 challenge. For CNN-ALEXNET and CNN-VGG models, the last 1000-unit soft-max layer (originally designed to predict 1000 classes) was replaced by a 2-unit soft max layer, which assigns a score for true and fake classes. The pre-trained model was further trained with the fingerprint datasets. The algorithm used to train CNN-Alex net and CNN. Figure 4 Illustration of a sequence of operations performed by a single layerconvolutional network in a sample image.is the Stochastic Gradient Descent (SGD) with a mini batch of size 5, using momentum [32] [33] 0.9 and a fixed learning rate of 10_{-6}. C. Local Binary Patterns Local Binary Patterns (LBP) are a local texture descriptor that have executed well in countless computer vision tenders, including texture classification and segmentation, image retrieval, surface inspection, and face detection [34]. It is a widely used method for fingerprint live ness detection [14] and it is used in this work as a baseline In its original version, the LBP operator assigns a label to every pixel of an image by thresholding each of the 8 neighbors of the 3x3-neighborhood with the center pixel value and considering the consequence as a unique 8-bit code representing the 256 possible neighborhood combinations. As the comparison with the neighborhood is performed with the central pixel, the LBP is an illumination invariant descriptor. The operator can be extended to use neighborhoods of different sizes [35]. Another extension to the original operator is the definition of so-called uniform patterns, which can be used to reduce the length of the feature vector and implement a simple rotation invariant descriptor [35]. An LBP is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. The number of different labels of LBP reduces from 256 to just 10 in the uniform pattern. The normalized histogram of the LBPs (with 256 and 10 bins for non-uniform and uniform operators, respectively) is used as a feature vector. The assumption underlying the computation of a histogram is that the distribution of patterns matters, but the exact spatial location does not. Thus, the advantage of extracting the histogram is the spatial invariance property. To investigate if location matters to our problem, we also implemented the method presented in [36], for face recognition, where the LBP filtered images are equally divided in rectangles and their histograms are concatenated to form a final feature vector. In this study, the histogram of the LBP image was further reduced using PCA, and a SVM with RBF kernel is used as the classifier. Similarly to the CNN-Random models, the hyper parameters, such as the number of PCA components and SVM regularization parameter, where found using an extensive brute force search on more than 2000 combinations, listed in table II. D. Increasing the Classifiers Generalization through Dataset Augmentation Dataset Augmentation is a technique that involves artificially creating slightly modified samples from the original ones. By using them during drill, it is likely that the classifier will converted more vigorous against small discrepancies that may be present in the data, forcing it to learn larger (and possibly more important) structures. It has been successfully used in computer vision benchmarks such as in [19], [37], and [38]. It is particularly suitable to out-of-core algorithms (algorithms that do not need all the data to be loaded in memory during training) such as CNNs trained with Stochastic Gradient Descent. Our dataset augmentation implementation is similar to the one presented in [19]: from each image of the dataset five smaller images with 80% of each dimension of the original images are extracted: four patches from each corner and one at the center. For each patch, horizontal reflections are created. As a result, we obtain a dataset that is 10 times larger than the original one: 5 times are due to translations and 2 times are due to reflections. At test time, the classifier makes a prediction by averaging the individual predictions on the ten patches.

3. Experiments:

A. Datasets: The datasets on condition that by the Live ness Finding Opposition (Liv Det) in the years of 2009 [39], 2011 [40], and 2013 [41] are used in this study. Liv Det 2009 comprises almost 18,000 images of real and fake fingerprints acquired from three different sensors (Biometrika FX2000, Cross match Verifier 300 LC, and Identix DFR 2100). Fake fingerprints were obtained from three different materials: Gelatin, Play Doh, and Silicone. Approximately one third of the images of the dataset are used for training and the remaining for

testing. Liv Det 2011 comprises 16,000 images acquired from four different sensors (Biometrika FX2000, Digital 4000B, Italdata ET10, and Sagem MSO300), each having 2000 images of fake and real fingerprints. Half of the dataset is secondhand for preparation and the other partial for testing. False fingerprints were obtained from four different materials: Gelatin, Wood Glue, Eco Flex, and Silgum. Liv Det 2013 comprises 16,000 images acquired from four different sensors (Biometrika FX2000, Cross match L SCAN GUARDIAN, Italdata ET10, and Swipe), each having approximately. 2,000 images of fake and real fingerprints. Partial of the dataset is castoff for keeping fit and the other half for taxing. Bogus impressions were obtained from five different materials: Gelatin, Latex, Eco Flex, Wood Glue, and Modasil. In all datasets, the factual/forged fingerprint ratio is 1/1 and they are equally disseminated between training and testing sets. The sizes of the images vary from sensor to sensor, ranging from 240x320 to 700x800 pixels, but they were all resized according to the input size of the pre-trained models, which is 224x224 for the CNN-Alexnet model and 227x227 pixels for the CNNVGG model.

B. Performance Metrics: The classification results were evaluated by the Average Classification Error (ACE), which is the standard metric for evaluation in Liv Det competitions. It is defined as ACE =SFPR + SFNR 2 (1) where SFPR (Spoof False Positive Rate) is the percentage of misclassified live fingerprints and SFNR (Spoof False Negative Rate) is the percentage of misclassified fake fingerprints.

C. Implementation Details: CNN-VGG and CNN-Random were trained using the Caffe package [42], which provides very fast CPU and GPU implementations and a user-friendly interface in Python. For the CNN-Random and LBP models, we wrote an improved cross-validation/grid search algorithm for choosing the best combination of hyper-parameters, in which each element of the pipeline is computed only when its training data is changed(the term element refers to operations such as preprocessing, feature extraction, dimensionality reduction or classification). This modification speeded-up the validation phase by approximately 10 times, although the gain can greatly vary as it depends on the element types and number of hyper parameters chosen. An important aspect of this work is that the algorithms. Run on cloud service computers, where the user can rent virtual computers and pay only for the hours that the machines are running. To train the algorithms, we used the GPU instances that allowed us to run dataset augmented experiments in a few hours; using traditional CPUs the training would take weeks.

4. Results:

The average error for each testing dataset is shown on Table III. Along with the models used in this study, we also show the error rate of the state-of-the-art method for each dataset, of which most of them were found in the compilation made by [43]. Particularly interesting results are for the Cross match 2013 dataset. As commented by [43], most techniques have problems in this dataset. For example, the LBP presents error rates close to zero at validation time and around 50% at test time. It container be seen from Liv Det 2013 rivalry results that this dataset is mainly hard to generalize, since nine of the eleven participants presented error rates greater than 45%. Contrary to these results, CNN models perform very well in this dataset, with error rates between 3.2%-4.7%.

Dataset State-of the-
Art
CNNVGG
CNNAlexnet
CNNRandom
LBP
Crossmatch 2013 7.9 [13] 3.4 4.7 3.2 49.4
Swipe 2013 2.8 [43] 3.7 4.3 7.6 3.3
Italdata 2013 0.8 [41] 0.4 0.5 2.4 2.3
Biometrika 2013 1.1 [17] 1.8 1.9 0.8 1.7
Italdata 2011 11.2 [43] 8.0 9.1 9.2 12.3
Biometrika 2011 4.9 [11] 5.2 5.6 8.2 8.8
Digital 2011 2.0 [43] 3.2 4.6 3.6 4.1
Sagem 2011 3.2 [11] 1.7 3.1 4.6 7.5
Biometrika 2009 1.0 [11] 4.1 5.6 9.2 10.4
Crossmatch 2009 3.3 [43] 0.6 1.1 1.7 3.6
Identix 2009 0.5 [43] 0.2 0.4 0.8 2.6
Average 3.5 2.9 3.7 4.7 9.6

A. Average Classification Error on Testing Datasets:

It is important to highlight that CNN-Random did require an exhaustive hyper-parameter fine tune (number of layers, filter size, number of filters, etc.) in order to get a model with good accuracy. On the other hand, the architectures of CNNA lexnet and CNN-VGG, which were already carefully selected for the Image Net object detection task, are general enough to be reused for the fingerprint Live ness detection task and yield excellent accuracy. Another interesting aspect is that the CNN-VGG performed better than the CNN-Alex net in

both object detection from ILSVRC-2012 and fingerprint Live ness detection tasks. This suggests that further improvements in models for object recognition can be applied to increase accuracy in fingerprint Live ness detection. The higher performance of our CNNVGG solution was confirmed as this model won the first place in the Fingerprint Live ness Detection 2015 Competition (Liv Det) 2015 [1], with an overall accuracy of 95.51%, while the second place achieved an overall accuracy of 93.23%. A. Effect of dataset augmentation Table IV compares the effect of dataset augmentation in our proposed models. Despite its longer training and running times, the technique helps to improve accuracy: the error was reduced by a factor of 2 in some cases. More importantly, the technique is not only effective on deep architectures, as commonly known, but also in shallow architectures, such as LBP.

Model No Augmentation With Augmentation
CNN-VGG 4.2 2.9
CNN-Alexnet 5.0 3.7
CNN-Random 9.4 4.7
LBP 21.2 9.6
Table IV
Augmentation Vs No Augmentation: Average Error on All Datasets.

B. Cross-dataset Evaluation: We would like to verify how a classifier would perform when unseen samples acquired from spoofy materials and individuals during training are presented at test time. Additionally, we want to test the hypothesis that the images share common characteristics for distinguishing fake fingerprints

Train Dataset Test Dataset CNN-VGG CNN-Alex net CNN-
Random LBP
Biometrika 2011 Biometrika 2013 15.5 15.9 20.4 16.5
Biometrika 2013 Biometrika 2011 46.8 47.0 48.0 47.9
Italdata 2011 Italdata 2013 14.6 15.8 21.0 10.6
Italdata 2013 Italdata 2011 46.0 49.1 46.8 50.0
Biometrika 2011 Italdata 2011 37.2 39.8 49.2 47.1
Italdata 2011 Biometrika 2011 31.0 33.9 46.5 49.4
Biometrika 2013 Italdata 2013 8.8 9.5 47.9 43.7
Italdata 2013 Biometrika 2013 2.3 3.9 48.9 48.4
Table V
AverageClassificationError on Cross-Dataset Experiments.
Dataset Materials - Train Materials - Test CNN-VGG CNNAlexnet CNN-Random LBP
Biometrika 2011 EcoFlex, Gelatine, Latex Silgum, Wood Glue 10.1 12.2 13.5 17.7
Biometrika 2013 Modalsil, Wood Glue EcoFlex, Gelatine, Latex 4.9 5.8 10.0 8.5
Italdata 2011 EcoFlex, Gelatine, Latex Silgum, Wood Glue, Other 22.1 25.8 26.0 30.9
Italdata 2013 Modalsil, Wood GlueEcoFlex, Gelatine, Latex 6.3
8.0 10.8 10.7

Average Classification Error on Cross-Material Experiments- from real ones, that is, the important features for classificationare independent from the acquisition device. For that, Cross dataset experiments were performed, which involve training a classifier using one dataset and testing on another. Forinstance, a cross-dataset experiment would involve training aclassifier using Biometrika-2011 dataset and testing it using Italdata-2013. In summary, these experiments should reflecthow well the classifier is able to learn relevant characteristics that distinguish real from fake fingerprints when samples acquiredfrom different environments and sensors are presented.We chose to use only Biometrika and Italdata sensors fromdatasets of years of 2011 and 2013 of the Liv Det competition, since executing all possible dataset combinations would bealmost impractical to run under the current computer architecture.All the models evaluated use dataset augmentation.Table V shows the testing error. CNN-Alex net and CNNVGGclearly outperform CNN-Random and LBP in mostcases. However, the testing error is still high (>20%) in 4out of 8 of the experiments, indicating that the models fail to generalize when the type of sensor used for testing is differentfrom the one used in training. Similarly, [14] reported thattheir multi-resolution LBP technique had poor results in cross deviceexperiments, with errors of around 40-50%.

C. Cross-Material Evaluation: Additionally to the influence of training and testing with different sensors (section IV-B), we investigated the performance of the classifiers when they are tested with spoofing materials never seen during training. The results are shown in Table VI. The error rates are lower than Cross-dataset experiments, which suggests that most of the generalization error can be attributed to different sensors and not to different materials.

D. Training All Datasets at Once: In this experiment we report the error rates when training and testing a single classifier using all datasets (2009, 2011, 2013), except for Swipe-2013 whose images are very different from the rest. The testing error rates, shown in Table VII, are compared with the results obtained when individual classifiers are trained per dataset, which are reported in Table III. The results show that training a

single classifier with all datasets yields comparable error rates when individual classifiers are trained per dataset, which suggests that the effort to design and deploy a Live ness detection system can be considerably reduced if all datasets are trained together, as the hyper parameter fine tuning needs to be performed for only one model. Model One Classifier trained with All Training

Datasets One Classifier per Dataset
CNN-VGG 3.4 2.9
CNN-Alexnet 4.1 3.7
CNN-Random 6.0 4.7
LBP 10.0 9.6 TABLE VII
Average Classification Error When a Single Classifier is Trained Using All Datasets Vs One Classifier per Dataset.

E. Pre-Training Effect: In this experiment the effect of using pre-trained networks is investigated. Table VIII compares the accuracy for the CNNVGG and CNN-Alex net models trained using only fingerprint images and when they are first pre-trained with the Image Net dataset and then fine tuned with fingerprint images. It can be seem that pre-training is necessary for those large networks as training them using only the fingerprint images results in over fitting.Model Training on Liv Det datasets Only Training on Image Net then Liv Det datasets CNN-VGG 49.4 (0.0) 2.9 (1.5)

CNN-Alexnet 48.1 (0.0) 3.7 (1.2)
Table VIII
Average ClassificationError For Testing Dataset Comparing the Efficacy of Pre-Trained Models with the Ones Solely Trained On The Liv Det Datasets.
The Training Error Is Showed In Parenthesis

F. Number of Training Samples vs Error: Deep learning techniques require large number of labeled training data in order to achieve a good performance when the models are initialized with random weights, since there are a lot of parameters that must be learned, thus requiring many samples. However, when the weights were already learned from another task, the number of required samples can be surprisingly low in order to achieve good accuracy. Figure 5 shows the number of training samples versus the average classification error in the test set for all datasets. Using only 400 training samples, CNN-VGG has almost the same performance as LBP using all the 18,800 training images. This suggests that less samples are needed when pre-trained models are used.

G. Processing Times: In real applications, a good fingerprint live ness detection system must be able to classify images in a short amount of time. Table IX shows the average testing/classification times for a single image (no augmentation) on a single core machine (1.8 GHz, 64-bit, with 4GB memory). We also show the times for training all datasets together. The pre-trained CNN models (CNN-Alex net and CNN-VGG) take around 5-40 hours to converge using a Nvidia GTX Titan GPU. The CNN-Random and LBP models take around 5-10 hours to converge on a 32-Cores machine (the larger portion of these times are required for dimensionality reduction using PCA).

Technique Training all Datasets Testing per Image (1-core-CPU)
CNN-VGG 20-40 hours (GPU) 650ms
CNN-Alex net 5-10 hours (GPU) 230ms
CNN-Random 5-10 hours (32-core CPU) 110ms
LBP 5-10 hours (32-core CPU) 50ms
Table IX
Average Training and Testing Times

5. Conclusion:

Convolutional Neural Networks were used to detect false vs real fingerprints. Pre-trained CNNs can yield state-of-the-art results on benchmark datasets without requiring architecture or hyper parameter selection. We also showed that these models have good accuracy on very small training sets (˜400 samples). Additionally, no task-specific hand-engineered technique was used as in classical computer vision approaches. Despite the differences between images acquired from different sensors, we show that training a single classifier using all datasets helps to improve accuracy and robustness. This suggests that the effort required to design a live ness detection system (such as hyper-parameters fine tuning) can be significantly reduced if different datasets (and acquiring devices) are combined during the training of a single classifier. Additionally, the pre-trained networks showed stronger generalization capabilities in cross-dataset experiments than CNN with random weights and the classic LBP pipeline. Dataset augmentation plays an important role in increasing accuracy and it is also simple to implement. We suggest that the method should always be considered for the training and prediction phases if time is not a major concern. Given the promising results provided by the technique, more types of image transformations should be included, such as color manipulation and multiple scales described in [44] and [45].

6. References:

1. V. Mura, L. Ghiani, G. L. Marcialis, F. Roli, D. A.Yambay, and S. A. Schuckers, "Liv Det 2015 fingerprint live ness detection competition 2015."
2. J. Galbally, F. Alonso-Fernandez, J. Fierrez, and J.Ortega-Garcia, "A high performance fingerprint Live ness detection method based on quality related features," Future Generation Computer Systems, vol. 28, no. 1, pp. 311–321, 2012.
3. Y. Chen, A. Jain, and S. Dass, "Fingerprint deformation for spoof detection," in Biometric Symposium, 2005, p. 21.
4. B. Tan and S. Schuckers, "Comparison of ridge and intensity-based perspiration live ness detection methods in fingerprint scanners," in Defense and Security Symposium. International Society for Optics and Photonics, 2006, pp. 62 020A–62 020A. [5] P. Coli, G. L. Marcialis, and F. Roli, "Fingerprint silicon replicas: static and dynamic features for vitality detection using an optical capture device," International Journal of Image and Graphics, vol. 8, no. 04, pp. 495–512, 2008. [6] P. D. Lapsley, J. A. Lee, D. F. Pare Jr, and N. Hoffman, "Anti-fraud biometric scanner that accurately detects blood flow," Apr. 7 1998, uS Patent 5,737,439.
5. A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni, "Fake finger detection by skin distortion analysis," Information Forensics and Security, IEEE Transactions on, vol. 1, no. 3, pp. 360–373, 2006.
6. D. Baldisserra, A. Franco, D. Maio, and D. Maltoni, "Fake fingerprint detection by odor analysis," in Advances in Biometrics. Springer, 2005, pp. 265–272.
7. A. K. Jain, Y. Chen, and M. Demirkus, "Pores and ridges: High-resolution fingerprint matching using level 3 features," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 29, no. 1, pp.15–27, 2007.
8. D. Gragnaniello, G. Poggi, C. Sansone, and L.Verdoliva, "Fingerprint Live ness detection based on weber local image descriptor," in Biometric Measurements and Systems for Security and Medical Applications (BIOMS), 2013 IEEE Workshop on. IEEE, 2013, pp. 46–50.
9. "Local contrast phase descriptor for fingerprint Live ness detection," Pattern Recognition, vol. 48, no. 4, pp. 1050–1058, 2015. [12] R. Frassetto Nogueira, R. de Alencar Lotufo, and R. Campos Machado, "Evaluating software-based fingerprint Live ness detection using convolutional networks and local binary patterns," in Biometric Measurements and Systems for Security and Medical Applications (BIOMS) Proceedings, 2014 IEEE Workshop on. IEEE, 2014, pp. 22–29.
10. D. Menotti, G. Chiachia, A. Pinto, W. Robson Schwartz, H. Pedrini, A. Xavier Falcao, and A. Rocha, "Deep representations for iris, face, and fingerprint spoofing detection," Information Forensics and Security, IEEE Transactions on, vol.
11. X. Jia, X. Yang, K. Cao, Y. Zang, N. Zhang, R.Dai, X. Zhu, and J. Tian, "Multi-scale local binary pattern with filters for spoof fingerprint detection," Information Sciences, vol. 268, pp. 91–102, 2014.
12. L. Ghiani, G. L. Marcialis, and F. Roli, "Fingerprint live ness detection by local phase quantization," in Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012, pp.