INTERNATIONAL JOURNAL OF SCIENTIFIC PROGRESS AND RESEARCH (IJSPR) ISSN: 2349-4689
Issue 99, Volume 34, Number 05, 2017

Biometric Liveness Authentication Detection

Sreejit Sundaran[1], Joycy K. Antony[2], Vipin K[3]

[1]*M.Tech Schola, Dept. of CSE, NCERC, Kerala*

[2]*Assistant Professor, Dept. of CSE, NCERC, Kerala*

[3]*Assistant Professor, Dept. of CSE, NCERC, Kerala*

Abstract— The active increase in demand and use of biometric authentication system recently, the spoofing of the same has increased. Because of this it has been increasingly important for detection of the live biometric from the fake ones. Biometric can be fingerprint, face, iris, voice, palm, or handwriting sign. Biometric technology has several advantages over common security methods based on some information. In this paper, using of the Local Binary Pattern (LBP) and convolutional neural networks (CNNs) for fingerprint liveness detection is done and as an extension to this additional face recognition in neural network is an added support to the Biometric liveness Authentication Detection.

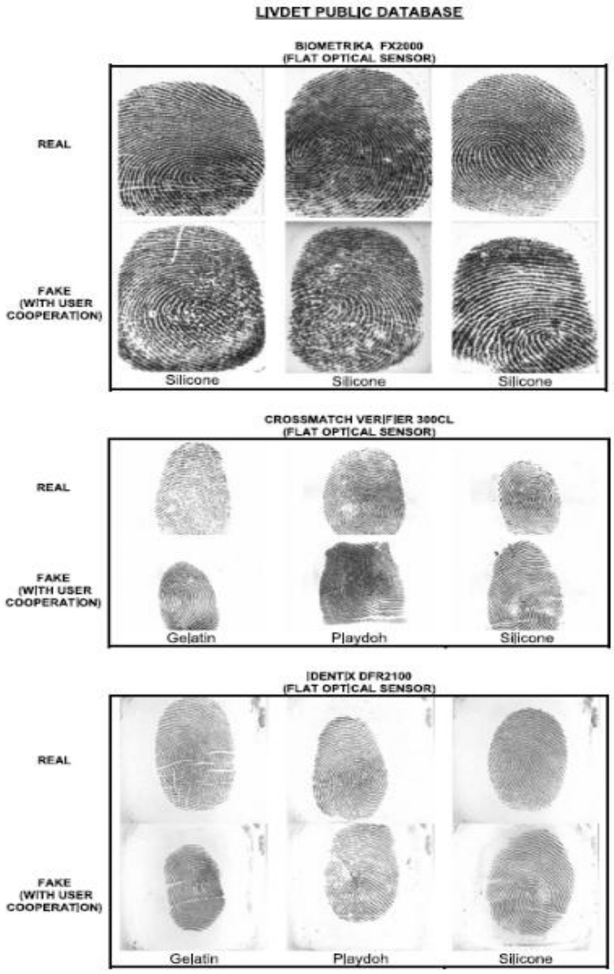
Keywords— Biometric Authentication, Neural Networks, Machine Learning.

I. INTRODUCTION

The biometrics is to automatically discriminate subjects in a reliable manner for a target application based on one or more signals derived from physical or behavioural traits, such as fingerprint, face, iris, voice, palm, or handwritten signature. Biometric technology presents several advantages over common security methods based on some information such as PIN, Password, etc. or physical devices such as key, card, etc. [2]. However, producing to the sensor device a fake physical biometric can be a simple way to overtake the systems security. Fingerprints, in specific, can be easily deceived from public materials, such as gelatine, silicone, and wood glue [2]. Therefore, a safe fingerprint system must correctly distinguish a spoof from a genuine finger (Figure 1). Different fingerprint liveness detection algorithms have been proposed [3]–[5], and they can be generally divided into two approaches: hardware and software.

In the hardware approach, a specific device is included to the sensor in order to detect particular abstracts 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 identified once the sample has been attained with a standard sensor. Fig. 1. Example of real and fake fingerprint images that can be obtained from the LivDet2009 database used in the experiments. Figure extracted from [9].

In the hardware approach, a specific device is included to the sensor in order to detect particular abstracts 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 identified once the sample has been attained with a standard sensor.



II. RELATED WORKS

The features used to extricate between actual and bogus fingers are extracted from the image of the fingerprint. There are methods such as those in [2] and [9], in which the features used in the classifier are based on specific fingerprint measurements, such as ridge strength, continuity, and clarity.

In contrast, some mechanism use general feature extractors such as Weber Local Descriptor (WLD) [10], which is a texture descriptor poised of differential excitation and orientation components. A new local descriptor that uses confined amplitude contrast (spatial domain) and phase (frequency domain) to produce a bi-dimensional contrast-phase histogram was proposed. In here two general feature extractors are compared: CNN with random weights, and Local Binary Patterns (LBP), whose multi-scale variant reported in attains good results in fingerprint liveness detection benchmarks. In contrast to more sophisticated systems that use texture descriptors as features vectors, such as Local Phase Quantization (LPQ), LBP with wavelets, and BSIF, their LBP application uses the original and uniform LBP coding schemes. Moreover, a variety of optional pre-processing methods such as contrast normalization, frequency filtering, and region of interest (ROI) extraction were attempted without success. Augmented datasets are effectively used to upsurge the classifiers robustness against small differences by generating additional samples from image translations and horizontal reflections. In this study we extend the work presented in by using a similar model from the well-known CNN.

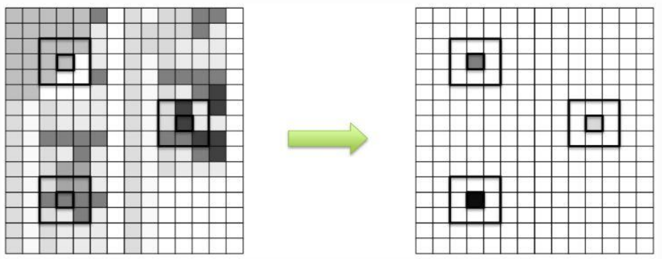
III. PROPOSED SYSTEM

Transfer learning is a research problem in machine learning that emphases on storing information gained while solving one problem and applying it to a different but related problem. In this study, we showed that it is possible to achieve state-of- the-art fingerprint liveness 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 ImageNET dataset (Figure 2), such as objects and scenes. In our study, fingerprint images were used, which differ significantly from those of other domains.

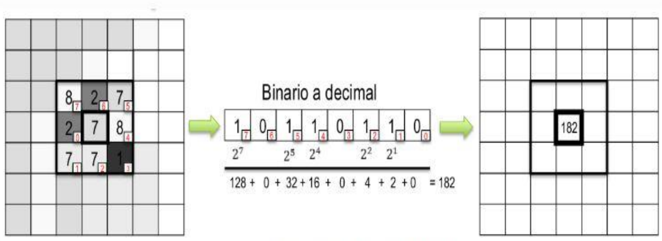
A. Local Binary Pattern

The first step in constructing the LBP texture descriptor is to convert the image to grayscale. For each pixel in the grayscale image, we select a neighbourhood of size r surrounding the center pixel. A LBP value is then calculated for this center pixel and stored in the output 2D array with the same width and height as the input image. For example, let’s take a look at the original LBP descriptor which operates on a fixed 3 x 3 neighbourhood of pixels just like this in the figure

We take the center pixel (highlighted in red) and threshold it against its neighbourhood of 8 pixels. If the intensity of the center pixel is greater-than-or-equal to its neighbour, then we set the value to 1; otherwise, we set it to 0. With 8 surrounding pixels, we have a total of 2 ^ 8 = 256 possible combinations of LBP codes.



To calculate the LBP value for the center pixel. We can start from any neighbouring pixel and work our way clockwise or counter-clockwise, but our ordering must be kept consistent for all pixels in our image and all images in our dataset. Given a 3 x 3 neighbourhood, we thus have 8 neighbours that we must perform a binary test on. The results of this binary test are stored in an 8-bit array, which we then convert to decimal, like this:



A primary benefit of this original LBP implementation is that we can capture extremely fine-grained details in the image.

B. Convolutional Networks

Convolutional Networks have demonstrated state-of-the-art performance in a range of image recognition benchmarks, such as MNIST, CIFAR-10, CIFAR-100, SVHN, and ImageNet. A classical CNN is composed of alternating layers of convolution and local pooling (i.e., subsampling). The aim of a convolutional layer is to take out patterns found within local regions of the entered images that are common throughout the dataset by convolving a template over the entered image pixels and displaying this as a feature map c, for each filter in the layer. The incentive behind pooling is that the activations in the pooled map *s* are less subtle to the precise locations of structures inside the image than the original feature map *c*. In a multi-layer model, the convolutional layers, which take the pooled maps as input, can thus extract features that are increasingly invariant to local transformations of the input image. This is important for classification tasks, since these transformations obfuscate the object identity. Attaining invariance to variations in position or lighting conditions, robustness to clutter, and compactness of representation, are all usual goals of pooling. Figure 4 demonstrates the feed-forward pass of a single layer convolutional network. The input sample is convoluted with three random filters of size 5 × 5 (engorged to make visualization easier), generating 3 convoluted images, which are then subject to non-linear function *max(x, 0)*, followed by a max-pooling operation, and subsampled by a factor of 2.

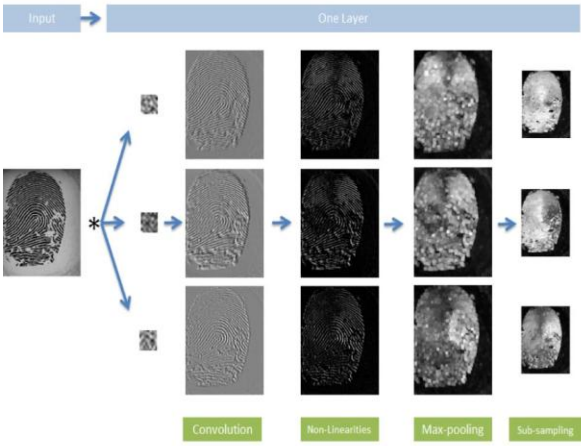
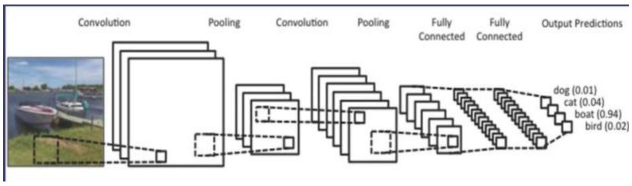
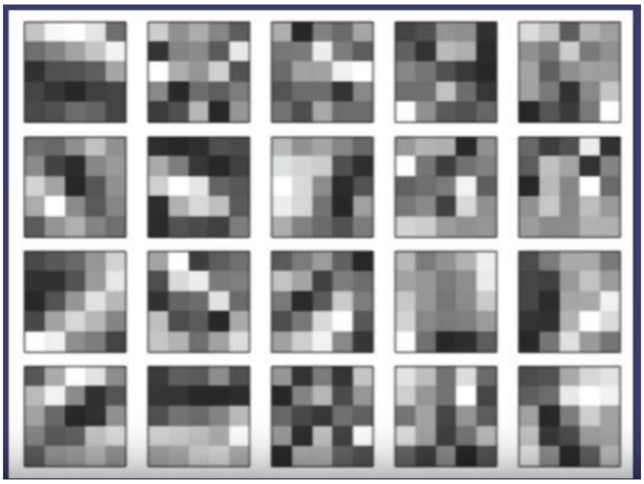


Fig. Illustration of a sequence of operations performed by a single layer convolutional network in a sample image.



CNN process flow diagram



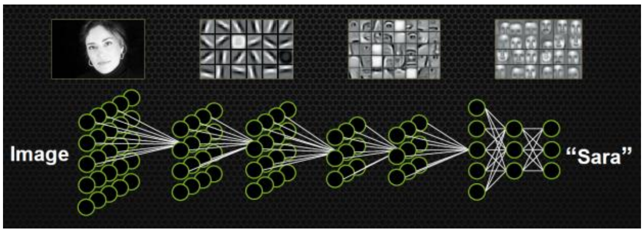
Hidden Layer output figure.

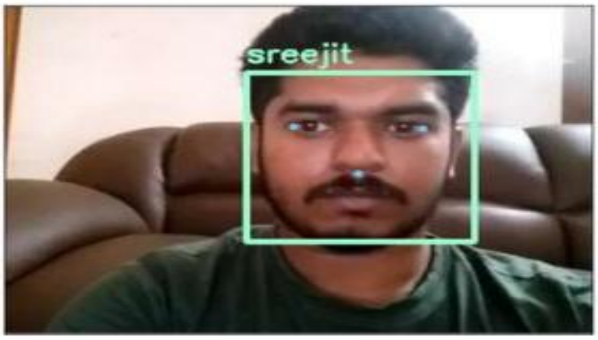
C. Face Recognition

It is a full face recognition pipeline on every frame, when a single person is trained, the classifier has no knowledge of other people and labels anybody with the name of the trained person. The web demo does not predict unknown users and the saved faces are only available for the browser session. If you're interested in predicting unknown people, one idea is to use a probabilistic classifier to predict confidence scores and then call the prediction unknown if the confidence is too low. This system is developed in the deep neural network.

The face are represented on a 128 dimensional unit hyper sphere. The following shows a 2d visualization of features using t-Distributed stochastic neighbor embedding (TSNE). At the start of the training phase the perimeter start of random. It is randomize as it learns.

The input layer is a face. Each of these hidden layers have some confidence level, and the average of that confidence level ends up being the entire confidence of the detected person.





D. Increasing the Classifier Generalization Through Dataset Augmentation

Dataset Augmentation is a technique that involves artificially creating slightly modified samples from the unique ones. By using them during training, it is expected that the classifier will become more robust against small differences that may be present in the data, forcing it to learn larger (and possibly more important) structures. It has been effectively used in computer vision benchmarks. It is particularly suitable to out-ofcore 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 trial time, the classifier makes a estimation by averaging the individual estimations on the ten patches.

E. Datasets

Bogus fingerprints were acquired from three different constituents: Gelatine, Play Doh, and Silicone. Roughly one third of the images of the dataset are used for training and the remaining for testing comprises 16,000 images acquired from four different sensors (Biometrika FX2000, Digital 4000B, Italdata ET10, and Sagem MSO300), each having 2000 images of bogus and real fingerprints. Half of the dataset is used for training and the other half for testing. Bogus fingerprints were obtained from four different materials: Gelatine, Wood Glue, Eco Flex, and Silgum. In all datasets, the real/fake fingerprint ratio is 1/1 and they are equally distributed between training and testing sets. The sizes of the images vary from sensor to sensor, ranging from 240×320 to 700×800 pixels, but they were all resized according to the input size of the pre-trained models, which is 227×227 pixels for the CNN-VGG model. And Iris datasets from Biometrica.

IV. EXPERIMENTAL RESULTS

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 close to zero at validation time and around 50% at test time. 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.

Table Iv: Augmentation Vs No Augmentation: Average Error On All Datasets

Model	No Augmentation	With Augmentation
CNN-VGG	4.2	2.9
CNN-Alexnet	5.0	3.7
CNN-Random	9.4	4.7

V. CONCLUSION

Convolutional Neural Networks were used to detect false vs real fingerprints and Iris. Pre-trained CNNs can yield state-of-the-art results on benchmark datasets without requiring architecture 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 liveness 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. Face recog

REFERENCES

[1] Rodrigo Frassetto Nogueira, Roberto de Alencar Lotufo, and Rubens Campos Machado, “Fingerprint Liveness Detection using Convolutional Networks,” in *Proc IEEE transactions on information forensics and security, vol. 11, no. 6, June 2016.*

[2] J. Galbally, F. Alonso-Fernandez, J. Fierrez, and J. Ortega-Garcia, “A high performance fingerprint liveness detection method based on quality related features,” *Future Generat. Comput. Syst.*, vol. 28, no. 1, pp. 311–321, 2012.

[3] Y. Chen, A. Jain, and S. Dass, “Fingerprint deformation for spoof detection,” in *Proc. Biometric Symp.*, 2005, p. 21.

[4] B. Tan and S. Schuckers, “Comparison of ridge- and intensity-based perspiration liveness detection methods in fingerprint scanners,” *Proc. SPIE*, vol. 6202, p. 62020A, Apr. 2006.

[5] P. Coli, G. L. Marcialis, and F. Roli, “Fingerprint silicon replicas: Static and dynamic features for vitality detection using an optical capture device,” *Int. J. Image Graph.*, vol. 8, no. 4, 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,” U.S. Patent 5 737 439, Apr. 7, 1998.

[7] A. Antonelli, R. Cappelli, D. Maio, and D. Maltoni, “Fake finger detection by skin distortion analysis,” *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 3, pp. 360–373, Sep. 2006.

[8] D. Baldisserra, A. Franco, D. Maio, and D. Maltoni, “Fake fingerprint detection by odor analysis,” in *Advances in Biometrics*. Heidelberg, Germany: Springer, 2005, pp. 265–272

[9] A. K. Jain, Y. Chen, and M. Demirkus, “Pores and ridges: Highresolution fingerprint matching using level 3 features,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 15–27, Jan. 2007.

[10] V. Mura, L. Ghiani, G. L. Marcialis, F. Roli, D. A. Yambay, and S. A. Schuckers, “Livdet 2015 fingerprint liveness detection competition 2015,” in *Proc. IEEE 7th Int. Conf. Biometrics Theory, Appl. Syst.*, Sep. 2015,pp.1–6.