2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Palmprint recognition via discriminative index learning

Jan Svoboda*, Jonathan Masci* and Michael M. Bronstein*†
*Institute of Computational Science, University of Lugano, Switzerland
†Perceptual Computing, Intel, Israel
Email: {jan.svoboda,jonathan.masci,michael.bronstein}@usi.ch

***Abstract*—In the past years, deep convolutional neural networks (CNNs) have become extremely popular in the computer vision and pattern recognition community. The computational power of modern processors, efficient stochastic optimization algorithms, and large amounts of training data allowed training complex tasks-specific features directly from the data in an end-to-end fashion, as opposed to the traditional way of using hand-crafted feature descriptors. CNNs are currently state-of-the-art methods in many computer vision problems, and have been successfully used in biometric applications such as face, fingerpring, and voice recognition. In palmprint recognition applications, CNNs have not yet been explored, and the majority of methods still rely on hand-crafted representations which do not scale well to large datasets and that usually require a complex manual parameter tuning.**

In this work, we show that CNNs can be successfully used for palmprint recognition. The training of our network uses a novel loss function related to the d-prime index, which allows to achieve a better genuine/impostor score distribution separation than previous approaches with only little training data required. Our approach does not require cumbersome parameter tuning and achieves state-of-the-art results on the standard IIT Delhi and CASIA palmprint datasets.

I. Introduction

Computer vision and pattern recognition research has been recently profoundly influenced by the deep learning paradigm. Deep learning is a powerful tool allowing to learn better tasks-specific data representations [1], [2], [3], [4], [5]. Its success was especially dramatic in image classification [1], [6], [7], shape analysis [5], [8], [9], [10], biometrics [11], [12], [13], [14] and many others. While sometimes criticized in the past for being "black boxes", there has recently been a significant progress in understanding and interpretation of what kind of features deep architectures are capable of learning [15], [16].

The modern convolutional neural network (CNN) architecture was introduced by LeCun *et al.* [17] in 1998. Unfortunately, the limited computational power of the computers of that time as well as the lack of large-scale datasets did not allow training sufficiently complex models that could cope with hard computer vision problems, and thus the capabilities of this methodology have not been fully appreciated until recently. Another popular Long-short term memory (LSTM) architecture, has become widely used only very recently, though introduced in 1997 [18].

In the domain of biometrics, one tries to produce a data representation that achieves a good separation of the genuine and impostor score distributions. Many successful face and palmprint recognition methods employ classical image processing techniques, trying to extract features by means of image filters that emphasize the discriminative information and attenuate the noise. Metric learning techniques and similarity sensitive hashing techniques are used to create a compact feature vector [19], [20]. Such filters are typically hand-crafted, and in general it is hard or even impossible to determine how the choice of the filter would affect the genuine and impostor distributions.

In this work, we approach the palmprint recognition problem with a deep learning paradigm we refer to as d-prime CNN, allowing to learn optimal features for the genuine/impostor separation task. The parameters of our neural networks are learned to minimize a loss function related to the d-prime discrimination index. Differently from previous learning approaches, rather than working on the computed scores directly (e.g., the siamese loss minimizing the distances between genuine samples and maximizing the impostor distances), we approximate the genuine/impostor score distributions as normal distributions and maximize their separation, trying to achieve the largest difference between the means and smallest standard deviations. By considering the whole distributions, rather than individual scores themselves, our network learns more general representation of palmprints and performs better on new, previously unseen, subjects.

The rest of the paper is organized as follows. In Section 2, we review the current state-of-the-art methods in palmprint recognition as well as the use of deep learning in biometrics. Section 3 describes the proposed d-prime CNN approach. Section 4 presents evaluation of our approach on standard palmprint databases, comparing our d-prime CNN to some of the state-of-the-art methods. Finally, Section 5 concludes the paper.

II. Background

A. Palmprint biometrics

Hand recognition is a well-established subfield of biometrics and a rather successful modality widely deployed commercially. Palmprint recognition makes use of the central area of a hand palm, which contains numerous wrinkles and creases. In the last decade, palmprint recognition has attracted an increasing attention, which resulted in the development of various palmprint analysis methods and rapid increase in performance.

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

Broadly speaking, we can characterize palmprint recognition techniques into five different categories [21]. *Ridge-based approaches* [22], [23] are based on analysis of the pattern of the ridges, position of the delta points, and the location of the minutiae. *Line-based approaches* [24], [25] typically perform edge detection followed by edge descriptors. *Subspace-based approaches* [26], [27], [28] make use of methods such as PCA, LDA, or ICA. *Statistical approaches* perform image analysis using Zernike moments [29], Bank of Binarized Statistical Image Features (B-BSIF) [30], etc. *Coding-* or *texture-based approaches* include PalmCode [31], FusionCode [32], Competitive Code [33], Ordinal Code [34], or Contour Code [35], which are currently among the state-of-the-art methods. Apart from these five main categories, there are also *hybrid approaches*, which combine several methods.

B. Deep learning for biometrics

Deep learning methods have been recently explored in the domain of biometrics. In particular, such approaches are widely used for face recognition [11], [12], where they outperformed all the previous models, including the very challenging settings of face recognition in the wild [36], [37]. An overview of CNN architectures used for face recognition can be found in [36]. CNNs have also been successfully employed for fingerprint [13], [38] or voice recognition [14].

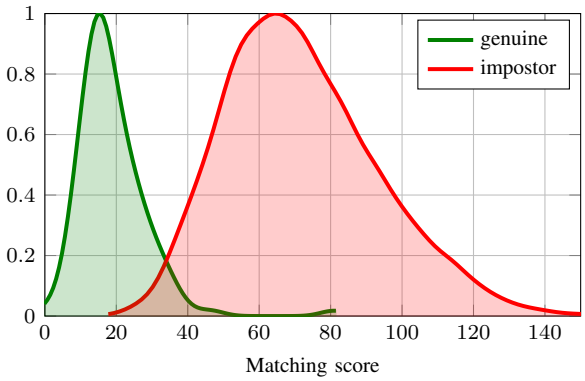
For palmprint recognition, we are only aware of two previous works. Jalali *et al.* [39] aim at contactless acquisition and claim to be deformation invariant; nevertheless, the authors did not evaluate their method on any of the standard contactless datasets. Minaee *et al.* [40] employed a Scattering CNN, which uses a static bank of wavelets and therefore no learning is involved. The authors show a good performance on the standard PolyU dataset, which is however captured in a touch-based manner. To the best of our knowledge, the approach proposed in this paper is the first use of CNNs for contactless palmprint recognition.

III. Our Method

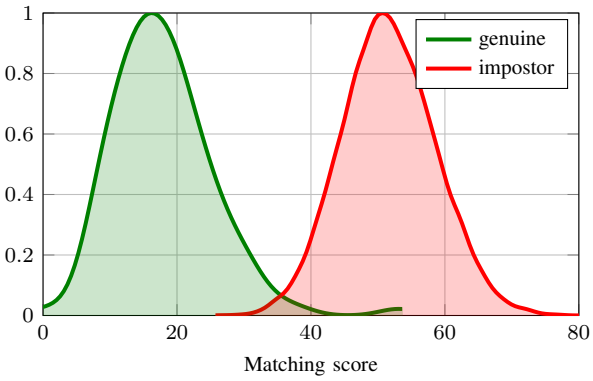
A. CNN architecture

We use a CNN applied to a 128×128 image of the palmprint region, extracted using a method such as [41], [42]. The CNN consists of a sequence of convolutional layers applying banks of filters to the input image or the output of the previous layer, pooling (non-linear averaging and downsampling), and linear combination. The weights of the filters are learnable parameters of the network, selected by an optimization procedure minimizing a task-specific loss function (in our case, a criterion of genuine/impostor separation defined in details below) on a training set.

The architecture of our network is illustrated in Fig. 2 comprises four convolutional, one pooling, one fully connected, and an output layer of 32 units. The output of the CNN is thus a 32-dimensional feature vector describing the input palrmprint image. The first two convolutional layers are simplified version of the first two layers of AlexNet [1]. After the first convolutional layer with rectified linear unit (ReLU) nonlinearity, we employ batch normalization [6] in order to re-normalize the input to the zero mean and unit variance, which improves convergence; all other layers do not use batch normalization as it did not further improve performance nor convergence.



(a) Siamese loss



(b) d-prime loss

Fig. 1. Score distributions produced with a CNN trained using the classical Siamese (a) and the proposed d-prime loss (b). Notice the better separation of the genuine/impostor distributions with the new loss.

The second convolutional layer has ReLU nonlinearity as well and is followed by a max pooling layer [43]. As pointed out by Yi *et al.* [11] in his work on face recognition, ReLU nonlinearity followed by max-pooling layer tends to learn sparse representations and discard information which leads to decrease in performance in some settings. During our experiments, we have observed the same behavior. Therefore, the remaining two convolutional layers have scaled $\tanh$ nonlinearity [44], [45] and are not followed by any pooling.

The output of the last convolutional layer is then fed to the linear fully connected layer with dropout of 30%. This layer is then connected to the last fully connected layer with 32 linear output units. Therefore, our method produces feature vectors of 32 values with 32-bit floating point precision, which is 128 byte in total.

B. d-prime loss

The key goal of biometric recognition systems is achieving a separation of the genuine and impostor score distributions.

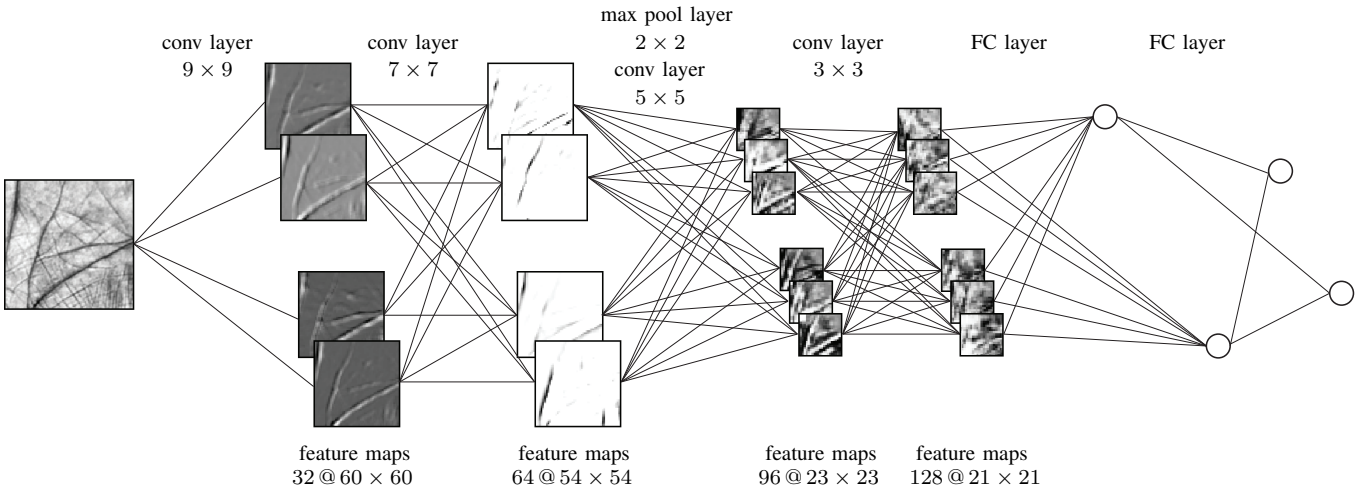


Fig. 2. Our CNN architecture, containing four convolutional layers. The first layer has a stride of 2 and a ReLU nonlinearity. The second has a stride of 1, ReLU, and it is followed by max pooling. The last two convolutional layers have strides of 1 and scaled $\tanh$ nonlinearity. The convolutional layers are followed by two fully connected layers, outputting a 32-dimensional feature vector.

A popular criterion is *d-prime* (also known as *sensitivity*- or *discriminative index*) [46], modeling the genuine and impostor score distributions as normal distributions $\mathcal{N}(\mu_{\text{gen}}, \sigma_{\text{gen}})$ and $\mathcal{N}(\mu_{\text{imp}}, \sigma_{\text{imp}})$ respectively, and measuring their separation as

$$d' = \frac{\mu_{\text{imp}} - \mu_{\text{gen}}}{\sqrt{\frac{1}{2}(\sigma^2_{\text{imp}} + \sigma^2_{\text{gen}})}}. \quad (1)$$

Well-separated distributions in the d' sense should thus have distant means and small variances. In this paper, we use a novel similar criterion of separation referred to as *d-prime*,

$$\ell = \sigma_{\text{gen}} + \sigma_{\text{imp}} + \mu_{\text{gen}} + \max\{0, M - \mu_{\text{imp}}\}, \quad (2)$$

where the last term is a standard hinge loss trying to pull the genuine/impostor means at least M apart.

Fig. 1 shows the genuine/impostor score distributions produced by our CNN using the proposed loss function (b), compared to the traditional *siamese loss* (a) [47], [48] which is applied to individual scores rather than to distributions

$$\ell_{\text{siam}} = \frac{1}{2}\|\mathbf{x} - \mathbf{x}^+\|^2 + \max\{0, M - \|\mathbf{x} - \mathbf{x}^-\|\}, \quad (3)$$

where $\mathbf{x}, \mathbf{x}^+$ and $\mathbf{x}^-$ denote the feature vectors of a subject, a genuine, and an impostor, respectively. Our d-prime loss results in a better separation of the genuine/impostor score distributions.

Given that samples are approximated by normal distributions, d-prime loss provides better generalization over training data and allows to train well having only little training dataset available.

We also have to derive the gradient of the introduced d-prime loss, which is necessary since we use AdaDelta gradient descent optimization algorithm. The gradient of the d-prime loss is as follows:

$$\frac{\partial \ell}{\partial d^i_{gen}} = \frac{1}{N}\left[2\left(d^i_{gen} - \mu_{gen}\right) + 1\right] \quad (4)$$

$$\frac{\partial \ell}{\partial d^i_{imp}} = \frac{1}{N}\left[2\left(d^i_{imp} - \mu_{imp}\right) - J\left(\mu_{imp} < M\right)\right], \quad (5)$$

where $\mu_{gen} = (\sum_{i=1}^N d^i_{gen})/N$, $\mu_{imp} = (\sum_{i=1}^N d^i_{imp})/N$, $\sigma^2_{gen} = \sum_{i=1}^N (d^i_{gen} - \mu_{gen})^2/N$, $\sigma^2_{imp} = \sum_{i=1}^N (d^i_{imp} - \mu_{imp})^2/N$ are the means and standard deviations of the normal distributions and $d^i_{gen} = \|x^i - x^i_+\|$ and $d^i_{imp} = \|x^i - x^i_-\|$ are the genuine and impostor distances respectively. N is the number of samples in a training batch and J is full unit matrix.

C. Learning

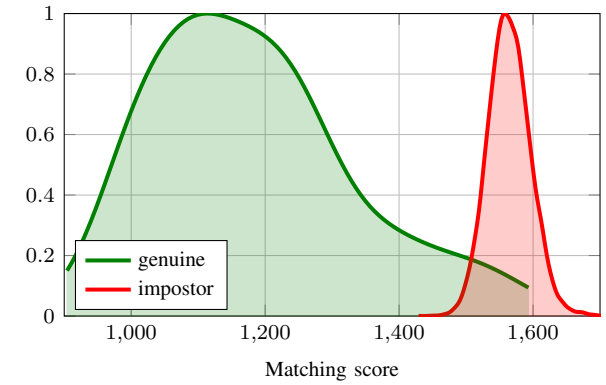
The training process used in our experiments consisted of 400 epochs in total. In each epoch, we processes 30 batches of 128 triplets of form (reference sample, positive sample, negative sample). We employ AdaDelta [49] updates with $\rho = 0.95$ and weight decay using L_2 regularization with $\mu = 10^{-4}$. The initial learning rate was set to 0.01 and was decreased each 100 epochs, consecutively to 5×10^{-3}, 10^{-3} and 5×10^{-4}.

Previous palmprint recognition works perturb the feature vectors while computing matching score in order to achieve translation invariance. We incorporate translation and scale invariance directly into our network during learning process by performing data augmentation. In particular, each training sample (palmprint image) is randomly shifted by a few pixels or slightly scaled (up to the difference of 8 pixels).

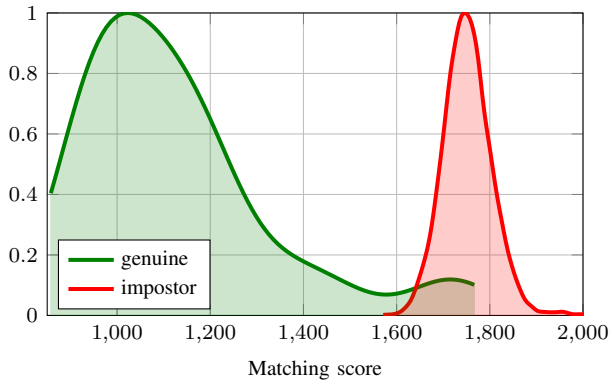
IV. Experiments

A. Data

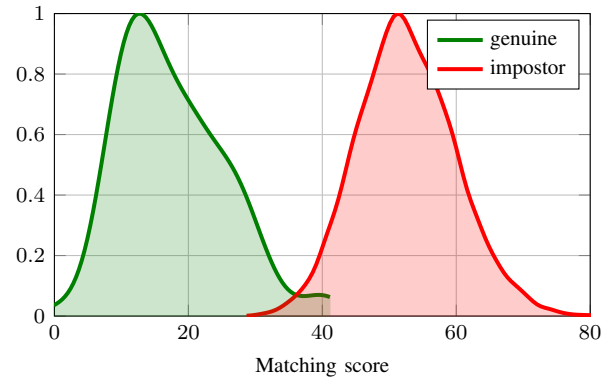
We tested the proposed model on two standard contactless palmprint datasets. The *IIT Delhi database* [50] contains 5 samples of segmented palmprint images for each of left and right hands of 230 different subjects. We used only right hand samples, and split the dataset into disjoint equally-sized training and test datasets, containing different subjects. We performed two-fold cross validation, always having 50% of subjects for training and 50% for testing. This process was



(a) Competitive Code



(b) Ordinal Code



(c) d-prime CNN

Fig. 3. Score distributions on the IIT Delhi dataset for two baseline approaches (a,b) and the proposed method (c). Genuine and impostor distributions are shown in green and red, respectively. Higher score corresponds to smaller similarity between the palmprint images. Our d-prime CNN achieves a better separation of the genuine/impostor score distributions.

repeated 8 times having different split of the subjects in each iteration (in each iteration, we have 5 samples for each subject and therefore we perform five times leave-one-out cross validation and average the results to get the performance for the current iteration).

The *CASIA database* [51] contains 5,502 images captured from 312 different subjects with approximately 9 images per subject (both left and right hands). The dataset did not provide

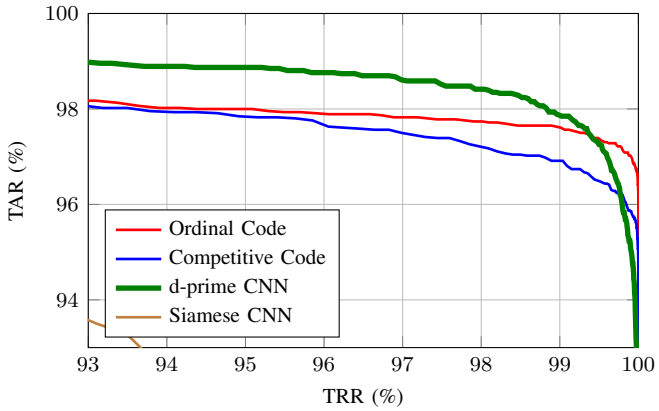


Fig. 4. ROC curve (tradeoff between acceptance and rejection rates) of the evaluated methods on the IIT Delhi dataset.

TABLE I
PERFORMANCE IN TERMS OF EER (THE LOWER THE BETTER).

Method	IIT Delhi	CASIA	Feature size
Competitive Code	2.33%	2.90%	384 Bytes
Ordinal Code	2.08%	2.41%	384 Bytes
Siamese CNN	6.08%	3.15%	128 Bytes
d-prime CNN	1.64%	1.86%	128 Bytes

extracted palmprint regions; we used a subset of subjects for which at least in 5 images the palmprint region was extracted successfully. The final dataset included 283 subjects for the right and 282 subjects for the left hand, always with 5 images per subject. The splitting and cross validation procedure was done in the same way as for the IITD dataset.

In all the experiments, our CNN was applied to palmprint images resized to a fixed 128×128 resolution. For comparison, we provide the results of two state-of-the-art methods, Ordinal Code and Competitive Code [52] using the code provided by the authors. All the methods were compared in terms of the ROC curves (tradeoff between acceptance and rejection rates), equal error rate (EER), and d-prime index.

B. Evaluation

Tables I and Table II summarize the performance of different methods on the IIT Delhi and CASIA datasets. Our d-prime CNN method outperforms the compared approaches in terms of EER and performs well also in terms of the d-prime index. The output feature vector size of our approach is 128 bytes, the smallest amongst the compared methods. Figs. 4 and 5 show the ROC curves of different methods. Our method achieves significantly lower false acceptance rates, which indicates its potential in large-scale applications. Figs. 3 depicts the genuine/impostor score distributions, speaking clearly in favor of our approach.

C. Feature visualization

In order to show what kind of information the output feature vectors represent, we followed Zeiler's visualization [53] in

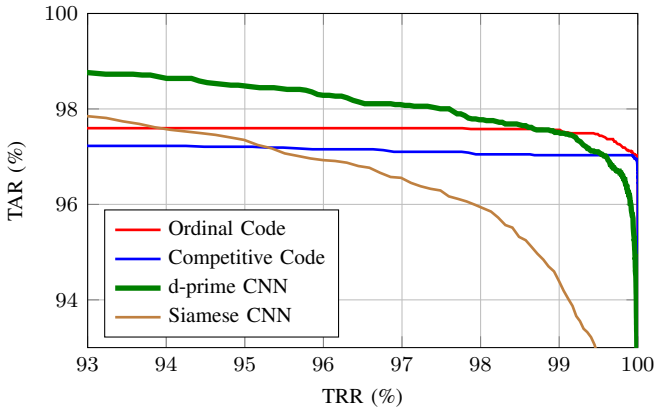


Fig. 5. ROC curve (tradeoff between acceptance and rejection rates) of the evaluated methods on the CASIA palmprint dataset.

TABLE II
GENUINE/IMPOSTOR DISTRIBUTIONS SEPARABILITY IN TERMS OF d-PRIME INDEX (THE HIGHER THE BETTER).

Method	IIT Delhi	CASIA
Competitive Code	3.32	4.72
Ordinal Code	3.92	5.75
Siamese CNN	2.91	2.20
d-prime CNN	4.84	5.73

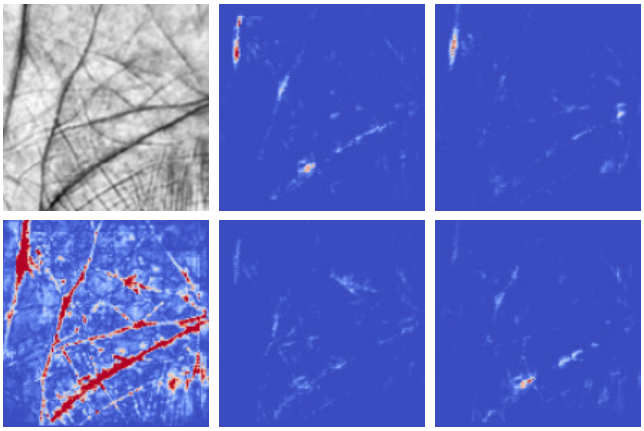


Fig. 6. Representation of an input image by the output feature vector of our CNN. The input image is in the top row on the left, the whole representation by all the features is in the bottom row on the left (red color indicates the relative importance of the pixel). The remaining four images on the right are representation of the input image by four random dimensions of the output feature vector.

our d-prime CNN in order to project features back through the network to obtain the representation of the input image mapped by a particular feature. We perform this evaluation to visualize that our network is learning meaningful information about the distinctive parts of the palmprint image. The resulting representation and several individual dimensions of an output feature vector are visualized in Fig. 6.

D. Cross-dataset learning

We have briefly investigated the possibility of learning the model on one dataset and applying it on another, which would be the ideal case. From our observations, there is a very good potential of learning model that generalizes across different datasets while performing regularization and augmentation more carefully. In general, palmprint images across different datasets have different contrast, brightness, sharpness, quality of alignment, etc. Taking all these factors into account, we observed that one can learn very good general model. We however do not evaluate this further in the paper.

V. CONCLUSION

In this paper, we explored the use of deep learning techniques for biometric palmprint image recognition. We presented a novel siamese-type convolutional neural network architecture, which is designed specifically for contactless palmprints. Unlike previous methods based on 'handcrafted' features requiring manual tuning, our approach automatically learns the features from the data and does not need cumbersome parameters tuning. Instead of training the network with the traditional siamese loss (trying to maximize the distance between positive and negative samples), we introduced a novel d-prime loss, which aims to maximize the separation of the genuine and impostor score distributions. We showed that our approach tends to learn features that provide better genuine/impostor score distributions separation, resulting in an improved recognition performance and better scalability.

One of the advantage of our approach in comparison to the non-learning counterparts is its simple transfer across different datasets. The network architecture stays the same and it is only necessary to retrain the filters. This way, one can obtain very good classifier for any palmprint database without any need to tune the parameters. Given the encouraging results (on par or outperforming several state-of-the-art approaches on standard benchmarks), we believe it is worthwhile to further explore palmprint recognition using convolutional neural networks.

ACKNOWLEDGMENT

We acknowledge the Chinese Academy of Sciences Institute of Automation for the CASIA Palmprint Database. The work is supported by the ERC Starting grant No. 307047 and the Swiss KTI/CTI (Federal Commission for Technology and Innovation) grant.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012.
[2] G. Papandreou, L.-C. Chen, and A. Yuille, "Modeling image patches with a generic dictionary of mini-epitomes," in *Proc. CVPR*, 2014.
[3] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Proc. NIPS*, 2012.
[4] H. Su *et al.*, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. ICCV*, 2015.
[5] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on riemannian manifolds," *arXiv:1501.06297*, 2015.

[6] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv:1512.03385*, 2015.

[8] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers, "Anisotropic diffusion descriptors," *Computer Graphics Forum*, vol. 35, no. 2, 2016.

[9] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst, "Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks," *Computer Graphics Forum*, vol. 34, no. 5, pp. 13–23, 2015.

[10] Z. Wu *et al.*, "3D ShapeNets: A deep representation for volumetric shape modeling," in *Proc. CVPR*, 2015.

[11] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv:1411.7923*, 2014.

[12] Y. Taigman, M. Yang, and M. R. dand L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. CVPR*, 2014.

[13] R. Wang, C. Han, Y. Wu, and T. Guo, "Fingerprint classification based on depth neural network," *arXiv:1409.5188*, 2014.

[14] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[15] K. Lenc and A. Vedaldi, "Understanding image representations by measuring their equivariance and equivalence," in *Proc. CVPR*, 2015.

[16] R. K. Srivastava, J. Masci, F. J. Gomez, and J. Schmidhuber, "Understanding locally competitive networks," in *Proc. ICLR*, 2015.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] J. Masci, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro, "Sparse similarity-preserving hashing," in *Proc. ICLR*, 2014.

[20] J. Masci, M. M. Bronstein, A. Bronstein, and J. Schmidhuber, "Multimodal similarity-preserving hashing," *PAMI*, vol. 36, no. 4, pp. 824–830, 2014.

[21] A. Genovese, V. Piuri, and F. Scotti, *Touchless Palmprint Recognition Systems*. Springer, 2014.

[22] E. Liu, A. K. Jain, and J. Tian, "A coarse to fine minutiae-based latent palmprint matching," *PAMI*, vol. 35, no. 10, pp. 2307–2322, 2013.

[23] M. Laadjel, A. Bouridane, F. Kurugollu, O. Nibouche, and W. Yan, *Transactions on Data Hiding and Multimedia Security*. Springer, 2010, ch. Partial Palmprint Matching Using Invariant Local Minutiae Descriptors, pp. 1–17.

[24] O. Nibouche and J. Jiang, "Palmprint matching using feature points and {SVD} factorisation," *Digital Signal Processing*, vol. 23, no. 4, pp. 1154–1162, 2013.

[25] X. Wu, K. Wang, and D. Zhang, "HMMs based palmprint identification," *Proc. ICBA*, 2004.

[26] D.-S. Huang, W. Jia, and D. Zhang, "Palmprint verification based on principal lines," *Pattern Recognition*, vol. 41, no. 4, pp. 1316 – 1328, 2008.

[27] S. Ribarić and M. Marčetić, "Personal recognition based on the gabor features of colour palmprint images," in *Proc. MIPRO*, 2012, pp. 967–972.

[28] Y. Wang and Q. Ruan, "Kernel Fisher discriminant analysis for palmprint recognition," in *Proc. ICPR*, vol. 4, 2006, pp. 457–460.

[29] W. l. Yang and L. l. Wang, "Research of palmprint identification method using Zernike moment and neural network," in *Proc. ICNC*, 2010.

[30] R. Raghavendra and C. Busch, "Robust palmprint verification using sparse representation of binarized statistical features: A comprehensive study," in *Proc. IHMMSEC*, 2014.

[31] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online palmprint identification," *PAMI*, vol. 25, no. 9, pp. 1041–1050, 2003.

[32] A. Kong, D. Zhang, and M. Kamel, "Palmprint identification using feature-level fusion," *Pattern Recognition*, vol. 39, no. 3, pp. 478 – 487, 2006.

[33] A. W. K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *Proc. ICPR*, 2004.

[34] Z. Sun, T. Tan, Y. Wang, and S. Z. Li, "Ordinal palmprint representation for personal identification," in *Proc. CVPR*, 2005.

[35] Z. Khan, A. Mian, and Y. Hu, "Contour code: Robust and efficient multispectral palmprint encoding for human recognition," in *Proc. ICCV*, 2011.

[36] G. Hu, Y. Yang, D. Yi, J. Kittler, W. J. Christmas, S. Z. Li, and T. M. Hospedales, "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition," *arXiv:1504.02351*, 2015.

[37] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. CVPR*, 2014.

[38] K. Cao and A. K. Jain, "Latent orientation field estimation via convolutional neural network," in *Proc. ICB*, 2015.

[39] A. Jalali, R. Mallipeddi, and M. Lee, "Deformation invariant and contactless palmprint recognition using convolutional neural network," in *Proc. HAI*, 2015.

[40] S. Minaee and Y. Wang, "Palmprint recognition using deep scattering convolutional network," *arXiv:1603.09027*, 2016.

[41] K. Ito, T. Sato, S. Aoyama, S. Sakai, S. Yusa, and T. Aoki, "Palm region extraction for contactless palmprint recognition," in *Proc. ICB*, 2015.

[42] H. K. Kalluri, M. V. N. K. Prasad, and A. Agarwal, "Dynamic ROI extraction algorithm for palmprints," in *Proc. ICSI*, 2012.

[43] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Proc. ICANN*, 2010.

[44] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Effiicient backprop," in *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*. Springer-Verlag, 1998, pp. 9–50.

[45] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. ICANN*, 2011.

[46] J. Daugman, "Biometric decision landscapes," University of Cambridge, Technical Report No. 482, 2000.

[47] J. Bromley *et al.*, "Signature verification using a "Siamese" time delay neural network," in *Proc. NIPS*, 1994.

[48] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. CVPR*, 2006.

[49] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," *arXiv:1212.5701*, 2012.

[50] "Iit delhi touchless palmprint database." [Online]. Available: http://web.iitd.ac.in/ajaykr/Database_Palm.htm

[51] "CASIA palmprint image database." [Online]. Available: http://biometrics.idealtest.org

[52] A. M. Zohaib Khan and Y. Hu, "Contour code: Robust and efficient multispectral palmprint encoding for human recognition," *Proc. ICCV*, pp. 1935–1942, Nov 2011.

[53] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *arXiv:1311.2901*, 2013.