Vein-based Biometric Verification using Densely-connected Convolutional Autoencoder

Rıdvan Salih Kuzu, *Student Member, IEEE,* Emanuele Maiorana, *Senior Member, IEEE* and, Patrizio Campisi *Senior Member, IEEE*

***Abstract*—In this paper, we propose a vein-based biometric verification system relying on deep learning. A novel approach consisting of a convolutional neural network (CNN), trained in a supervised manner, cascaded with an auto-encoder, trained in an unsupervised way, is here exploited. In more detail, a novel densely-connected convolutional autoencoder is here used on top of backbone CNNs. This architecture aims at increasing the discriminative capability of the features generated from hand vein patterns. Experimental tests on finger, palm, and dorsal veins show that the proposed approach leads to an improvement of the recognition rates with respect to the use of the sole CNNs for feature extraction. The achieved performance are superior to the current state of the art in vein biometric verification.**

***Index Terms*—Biometric recognition, Vein patterns, Deep learning, Convolutional Neural Networks, Sparse Auto-encoders.**

I. Introduction

THE recent advances in deep learning are leading to tremendous changes also in the biometric framework [1]. In fact, supervised learning, using architectures such as convolutional neural networks (CNNs), has been widely adopted for feature extraction and template comparison, with application to several biometric traits [2]. Generative models, such as autoencoders and generative adversarial networks, have also been exploited to produce biometric representations, typically resorting to an unsupervised learning strategy [3].

The present study investigates, for the first time, how the exploitation of a cascade network, composed of both a supervised CNN and an unsupervised autoencoder, can increase the effectiveness of the feature extraction process, within the vein-based verification framework. A novel densely-connected convolutional autoencoder has been designed and used on top of backbone CNNs to improve the discriminative capability of the generated templates. The effectiveness of the proposed approach has been evaluated through extensive experimental tests carried out in the verification scenario, using vein patterns obtained from finger, palm, and the back of the hand.

In Section II, a brief overview of vein-based verification systems relying on deep learning approaches is given, with the emphasis on methods exploiting autoencoders. The proposed cascade network is described in Section III, where both the used backbone CNNs, as well as the proposed densely-connected convolutional autoencoder, are described. The experimental results are presented in Section IV, while conclusions are finally drawn in Section V.

R.S. Kuzu, E. Maiorana, P. Campisi, are with the Section of Applied Electronics, Department of Engineering, Roma Tre University, Via Vito Volterra 62, 00146, Rome, Italy e-mail: {ridvansalih.kuzu, emanuele.maiorana, patrizio.campisi}@uniroma3.it, Phone: +39.06.57337064, Fax: +39.06.5733.7026.

II. Vein based verification

Several approaches relying on deep learning techniques have been proposed within the framework of vein-based biometric verification. A summary is given in Table I, whereas for a detailed overview the reader is referred to [12].

The majority of the applications of deep learning to biometric recognition is based on the supervised use of discriminative models, such as CNNs or recurrent neural networks (RNNs), to extract characteristics peculiar to each subject from the available identifiers. To this aim, some works have recently exploited generative models, usually trained in an unsupervised manner. Within this framework, autoencoders have been typically employed to perform dimensionality reduction or denoising of the considered biometric samples. In fact, an autoencoder is meant to learn, from a given set of data, in an unsupervised manner, an efficient and compact representation, i.e., an encoding, usually projecting the original space into a lower-dimensional one [13]. In order to achieve such goal, two networks have to be optimized during the training stage: an encoder, mapping the input into a latent code, and a decoder, mapping the latent code into an approximate reconstruction of the original input. Typically, the learning process aims at forcing the generated representations of the input to assume some useful properties such as sparseness [14]. Representations of behavioural biometric traits learned through autoencoders have been used in the framework of keystroke data [15], online signatures [16], latent fingerprint [17], periocular biometrics [18], and face recognition [19], [20], [21], to cite just a few.

Some attempts to use autoencoders for vein-based recognition have also been performed. An approach to extract features from finger veins with a single-hidden-layer autoencoder, and performing comparisons using one-class Gaussian classifiers, has been proposed in [22]. Representations generated using convolutional autoencoders have been employed to perform identification with classifiers based on neural networks and support vector machines (SVMs) in [23] and [24], respectively. Siamese networks have also been trained for verification purposes in [25], where features generated by autoencoders have been used as input. However, all the aforementioned approaches could not achieve recognition performance as good as those obtained using CNNs to create vein pattern representations [4], [26]. Conversely, the method here proposed, and described in Section III, exploits autoencoders to generate templates characterized by an improved discriminative capability, with respect to the use of CNNs only, thus attaining recognition performance better than current state of the art.

TABLE I
DEEP-LEARNING-BASED VASCULAR BIOMETRIC VERIFICATION SYSTEMS: STATE-OF-THE-ART

Paper	Trait	Database			Proposed System		Performance
		Name	# of Subjects	Comments	Feature Extraction	Comparison	
[4]	Dorsal Vein Finger Vein Palm Vein Palm Vein	OWN FV-USM PUT PolyU-P	200 123 50 200	Training and test data belongs to different sessions. Subjects are not disjoint in the partitions.	Multi-weighted Co-occurence Descriptor Encoding	Large Margin Distribution Machine	EER=0.015% EER=0.307% EER=0.615% EER=0.017%
[5]	Palm Vein	PolyU-P CASIA	250 100	Training and test data belongs to different sessions. Subjects are not disjoint in the partitions.	Iterative Deep Belief Network	Hamming Distance	EER=0.015% EER=0.330%
[3]	Finger Vein	THU-FVFDT2 SDUMLA	610 106	Subjects in THU-FVFDT2 are split into disjoint sets for training, validation, and test. SDUMLA is employed for system verification	Generative Adversarial Networks for Finger Vein (FV-GAN)	Cross Correlation on Binarized Templates	EER=1.12% EER=0.94%
[6]	Palm Vein	PolyU-P CASIA IITI	250 100 185	The first half samples are utilized as the gallery and the remaining ones as the probe.	U-Net like Decoder-Encoder CNN Architecture	Siamese Matching Network	EER=0.66% EER=3.71% EER=0.93%
[7]	Finger Vein	SDUMLA PolyU-F	106 156	Subjects in each dataset are split into two equal size sets. 2-fold cross validation is made. The results on PolyU-F are biased due to pairing of data in the same session.	Densenet-161 on Composite Finger Images	Shift Matching and Minimum Rule	EER=2.35% EER=0.33%
[8]	Palm Vein Dorsal Vein	PolyU-P OWN	250 200	No information about training and test partitions.	Spatial Pyramid Pooling on Pre-trained VGG16	SVM Classifier	EER=0.068% EER=0.060%
[9]	Finger Vein	MMCBNU SDUMLA	100 106	Genuine and Impostor pairs for training and testing do not come from disjoint subjects. EER results are based on the average of 5 pairs	2-Channel Network, 2-Stream Network, and Selective Network based on Intra-Class Variations	SVM Classifier with Linear Kernel	EER=0.10% EER=0.47%
[10]	Finger Vein	SDUMLA UTFVP	106 60	Each dataset is divided into two disjoint subsets containing 720 samples for training and tests. Cross-testing among datasets is reported.	U-Net, RefineNet and SegNet on Human Annotated Pixel-Labels	Cross Correlation on Binarized Templates	EER=2.63% EER=0.64%
[11]	Finger Vein	PolyU-F	156	105 subjects are used for training and tests are conducted on the remaining 51 subjects.	Modified VGG-16 with Supervised Discrete Hashing	Hamming Distance	EER=9.77%

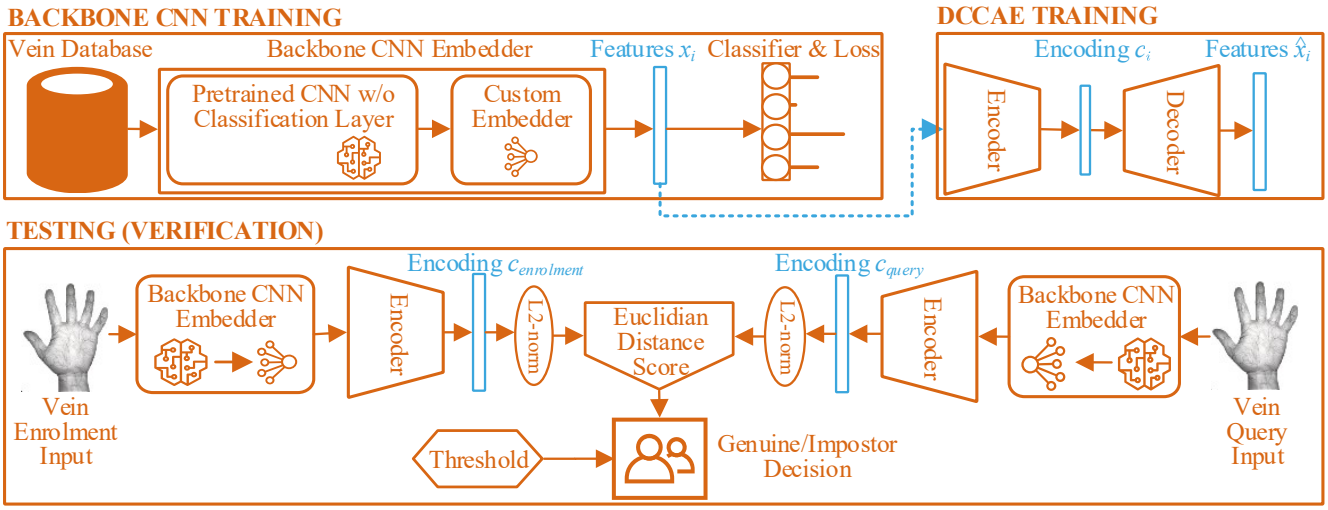

Fig. 1. High-level representation of the training (above) and testing (below) phases for the proposed system, based on a backbone CNN and a DCCAE.

III. PROPOSED SYSTEM

As outlined in Section II, autoencoders have been used so far in biometric recognition systems to learn features, in an unsupervised manner, directly from the employed traits.

In this paper, we take a different perspective as we evaluate the effectiveness of a novel approach, in which a supervised CNN is first trained on the original biometric data, and then an autoencoder is employed to refine the obtained representations and to improve the achievable recognition performance. Specifically, the proposed cascade network is illustrated in Figure 1. Feature learning is performed by sequentially training the following two branches:

(i) a backbone CNN, trained in a supervised classification mode. In the proposed implementations, following the approach adopted in [12] and [26], we employ CNNs derived from DenseNet-161 [27] and ResNext-101 (32x8d) [28], initialized with weights trained over the ImageNet database. As detailed in [12] and [26], the final fully-connected layers of the considered CNNs are replaced with a custom embedder, comprising a 7×7 average pooling layer with batch normalization, a 50% dropout, and a fully-connected layer with batch normalization producing representations with $d_{CNN} = 1024$ features. Training is performed using a loss based on cross-entropy with additive angular margin penalty (AAMP) [29];

(ii) an autoencoder, taking as input the representations generated by the employed backbone CNNs. The so obtained representations are refined by learning, in an unsupervised manner, a compact feature encoding with size $d_{AE} = 256$. In more detail, in this paper, we take an innovative approach using, for the first time in literature, a densely-connected convolutional autoencoder (DCCAE) for biometric recognition purposes.

It is worth remarking that, in the experimental tests we have conducted using cascade networks based on fully-connected and convolutional autoencoders, it has not been possible to achieve any performance improvement, with respect to the use of backbone CNNs only.

On the other hand, taking inspiration from the benefits stemming by the use of DenseNet architectures to perform vein-based recognition [26], resorting to dense-blocks when designing the proposed autoencoder has proved to be effective. Networks comprising dense blocks can be typically defined through a limited number of parameters, thanks to the concatenation of the output of each employed layer, thus alleviating the vanishing gradient problem and achieve better generalization during the training phase. Therefore, training can be efficiently carried out even with a small amount of data, as it may happen when dealing with biometric data.

The encoder and decoder components of the proposed DCCAE are summarized in Tables II and III, respectively. All

TABLE II
ENCODER COMPONENT OF THE PROPOSED DCCAE

h	Layer		Input Size	Output Size
1	Input Layer	1×3 Conv	1×1024	16×1024
2-6	Dense Block 1	[Batchnorm, ReLU 1×3 Conv] $\times 5$	16×1024	80×1024
7	Transition 1	Batchnorm, ReLU 1×3 Conv, str.2	80×1024	32×512
8-12	Dense Block 2	[Batchnorm, ReLU 1×3 Conv] $\times 5$	32×512	80×512
13	Transition 2	Batchnorm, ReLU 1×3 Conv, str.1	80×512	64×512
14-18	Dense Block 3	[Batchnorm, ReLU 1×3 Conv] $\times 5$	64×512	80×512
19	Transition 3	Batchnorm, ReLU 1×3 Conv, str.2	80×512	32×256
20-24	Dense Block 4	[Batchnorm, ReLU 1×3 Conv] $\times 5$	32×256	80×256
25	Transition 4	Batchnorm, ReLU 1×3 Conv str.2	80×256	16×128
26	Hidden Encoder	Fully-connected Layer Batchnorm, Sigmoid	1×2048	1×512
27	Latent Encoder	Fully-connected Layer Batchnorm, Sigmoid	1×512	1×256

TABLE III
DECODER COMPONENT OF THE PROPOSED DCCAE

h	Layer		Input Size	Output Size
28	Latent Decoder	Fully-connected Layer Batchnorm, Sigmoid	1×256	1×512
29	Hidden Decoder	Fully-connected Layer Batchnorm, Sigmoid	1×512	1×2048
30	Transition 1	Batchnorm, ReLU 1×3 Tran-conv, str.2	16×128	32×256
31-35	Dense Block 1	[Batchnorm, ReLU 1×3 Tran-conv] $\times 5$	32×256	80×256
36	Transition 2	Batchnorm, ReLU 1×3 Tran-conv, str.2	80×256	64×512
37-41	Dense Block 2	[Batchnorm, ReLU 1×3 Tran-conv] $\times 5$	64×512	80×512
42	Transition 3	Batchnorm, ReLU 1×3 Tran-conv, str.1	80×512	32×512
43-47	Dense Block 3	[Batchnorm, ReLU 1×3 Tran-conv] $\times 5$	32×512	80×512
48	Transition 4	Batchnorm, ReLU 1×3 Tran-conv, str.2	80×512	16×1024
49-53	Dense Block 4	[Batchnorm, ReLU 1×3 Tran-conv] $\times 5$	16×1024	80×1024
54	Transition 5	Batchnorm, ReLU 1×3 Tran-conv, str.2	80×1024	16×1024
55	Output Layer	1×3 Tran-conv	16×1024	1×1024

dense blocks employed in the employed architecture have a growth rate equal to 16, as shown in Figure 2. When training the proposed DCCAE, the loss to be minimized is defined as

$$Loss = L_R + \beta \cdot L_S, \tag{1}$$

being β the sparsity tuning parameter. Cosine dissimilarity is used for the reconstruction loss L_R, with

$$L_R = \frac{1}{B}\sum_{i=1}^{B}[1 - \cos(x_i, \hat{x}_i)], \tag{2}$$

being x_i the feature representation of a i-th vein sample produced by the backbone CNN feature embedder, the feature representation reconstructed by the decoder, and B the batch size. Three different losses L_S have been employed to evaluate the sparsity of the produced encodings, resorting to Kullback-Leibler divergence D_{KL}, $L1$ regularization, and to spectral restricted isometry property (SRIP) regularization [30]. The sparsity loss based on D_{KL} is defined as:

$$L_S^{D_{KL}} = \sum_{h=26}^{29}\sum_{j=1}^{N^{(h)}} D_{KL}(\rho \,||\, \hat{\rho}_j^{(h)}), \quad \hat{\rho}_j^{(h)} = \frac{1}{B}\sum_{i=1}^{B}[a_j^{(h)}(x_i)] \tag{3}$$

where $a_j^{(h)}$ is the j-th activation output of the h-th hidden layer when x_i is fed as input to the DCCAE, with $j = 1, \ldots, N^{(h)}$, being $N^{(h)}$ the number of activation units in the h-th hidden layer. This loss therefore depends on the divergence of the average activation of a neuron j in the hidden layer h from the uniform distribution defined by the sparsity parameter $\rho \in [0, 1]$. The $L1$ sparsity loss is instead defined as:

$$L_S^{L1} = \sum_{h=26}^{29}\sum_{j=1}^{N^{(h)}} |a_j^{(h)}|. \tag{4}$$

Both $L_S^{D_{KL}}$ and L_S^{L1} sparsity losses therefore depend on the activation outputs of four layers of the proposed DCCAE,

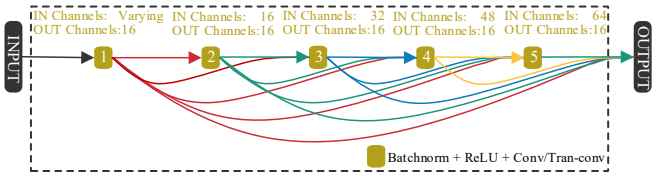

Fig. 2. Dense block with 5 Batchnorm-ReLU-Conv/Tran-conv dense layers.

namely the last two in the encoder and the first two in the decoder. Conversely, the loss based on SRIP regularization is computed on the weights of each convolutional layer of the proposed DCCAE, that is:

$$L_S^{SRIP} = \sum_{h=1}^{55} \sigma(W^{(h)\top} W^{(h)} - I), \tag{5}$$

being $W^{(h)}$ a matrix containing the weights of the h-th layer, σ the spectral norm which calculates the supremum of its argument as defined in [30], and I an identity matrix. Here, it is worth mentioning that, differently from the $L_S^{D_{KL}}$ and L_S^{L1} losses, L_S^{SRIP} has been originally defined for orthogonality regularization, instead of for generating sparse representations. Its effectiveness in generating discriminative templates from biometric data by means of autoencoders is here evaluated for the first time in literature.

IV. EXPERIMENTAL ANALYSIS

The proposed system, jointly employing CNN and DCCAE, has been tested over three different kinds of vascular traits, namely finger vein using the SDUMLA database [31], palm vein with the PolyU-P database [32], and dorsal vein through the Bosphorus database [33].

Open-set scenarios, in which the classes available in each dataset are split into two equally-sized disjoint partitions, one used for training with 20% of training data reserved for validation, and the other one employed for verification tests, have been considered. For databases comprising multi-condition (Bosphorus) and multi-session (PolyU-P) data, samples from different conditions/sessions have been employed for enrolment and verification purposes, to avoid any bias effect.

Backbone CNNs have been trained using stochastic gradient descent with momentum (SGDM) and a batch size of 32. The AAMP margin and scale have been selected in the ranges $[0.3, 0.7]$ and $[16, 96]$, respectively, and weight decay chosen within the set $\{0.1, 0.05, 0.001, 0.0005\}$. For the proposed DCCAE branch, Glorot uniform initialization is used for convolutional and fully-connected layers, and unit weight initialization for batch normalization. Training is performed using SGDM with a batch size of 128. The weight decay hyper-parameter

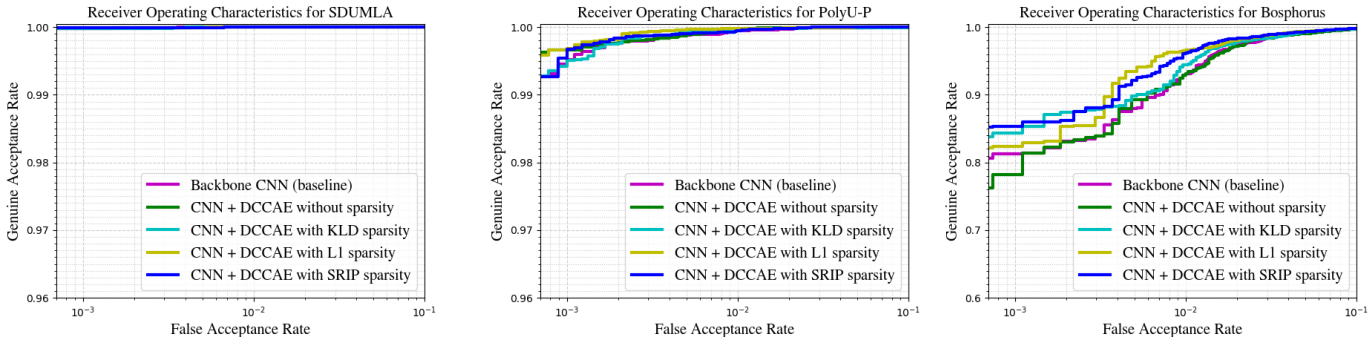

Fig. 3. ROCs of systems using DenseNet-161 as backbone CNN: SDUMLA (left), PolyU-P (center), Bosphorus (right).

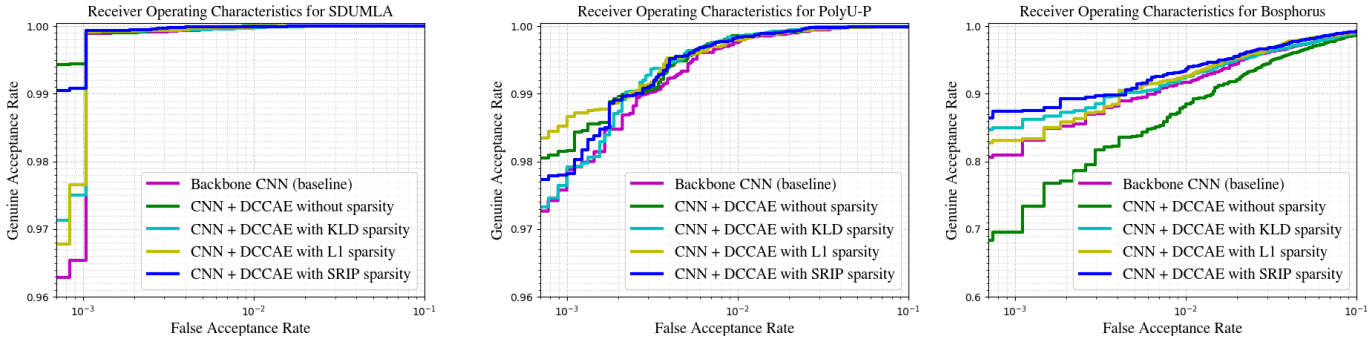

Fig. 4. ROCs of systems using ResNext-101 as backbone CNN: SDUMLA (left), PolyU-P (center), Bosphorus (right).

TABLE IV
RECOGNITION RATES (%) USING DENSENET-161 AS BACKBONE CNN.

Database	CNN only (no DCCAE)		CNN + DCCAE							
			$L_S^{D_{KL}}$		L_S^{L1}		L_S^{SRIP}		No sparsity	
	EER	GAR	EER	GAR	EER	GAR	EER	GAR	EER	GAR
SDUMLA	0.023	99.97	0.012	99.98	0.015	99.99	**0.009**	**99.99**	0.025	99.98
PolyU-P	0.222	99.52	0.194	99.51	0.194	**99.69**	**0.189**	99.68	0.228	99.67
Bosphorus	2.33	81.35	2.07	84.34	1.92	82.37	**1.78**	**85.33**	2.28	78.20

TABLE V
RECOGNITION RATES (%) USING RESNEXT-101 AS BACKBONE CNN.

Database	CNN only (no DCCAE)		CNN + DCCAE							
			$L_S^{D_{KL}}$		L_S^{L1}		L_S^{SRIP}		No sparsity	
	EER	GAR	EER	GAR	EER	GAR	EER	GAR	EER	GAR
SDUMLA	0.105	96.55	0.088	97.51	0.094	97.66	**0.085**	**99.09**	0.103	99.45
PolyU-P	0.528	97.88	0.483	97.92	0.467	**98.67**	**0.439**	97.82	0.506	98.15
Bosphorus	3.54	80.97	3.48	84.98	3.22	83.11	**3.18**	**87.45**	4.11	69.58

has been selected in the set $\{0.1, 0.05, 0.001, 0.0005\}$, while the sparsity tuning parameter β falls in the range $[0.00001, 25]$.

An ablation study has been carried out to highlight the effects and the relevance of using the proposed DCCAE in addition to a standalone CNN, as well as the effectiveness of different sparsity losses, by evaluating the performance achievable when exploiting:

- standalone CNNs without any sparsity loss (this scenario represents the current state of the art);
- a cascade network comprising a CNN and the proposed DCCAE (CNN + DCCAE), without any sparsity loss;
- a CNN + DCCAE network, with L1 sparsity loss;
- a CNN + DCCAE network, with KLD sparsity loss;
- a CNN + DCCAE network, with SRIP sparsity loss.

The obtained performance, in terms of receiver operating characteristics (ROCs), are shown in Figures 3 and 4, and summarized in terms of equal error rate (EER) and genuine acceptance rate (GAR) at false acceptance rate (FAR) equal to 10^{-3} in Tables IV and V, respectively. It is worth remarking that the recognition rates reported in Table IV, for the case where the proposed DCCAE is not used, currently represent the best performance achievable when using hand vein patterns [26]. The reported results show that the designed DCCAE is able to improve the performance achievable using only CNNs in all the considered scenarios, that is, for finger, palm, and dorsal vein, as well as for both the considered backbone CNNs. The obtained results show that the sparseness of the generated encodings affects the discriminative capability of the generated representations, with better performance typically achieved when using the SRIP regularization.

V. CONCLUSIONS

We have proposed a novel feature extraction pipeline for vein-based biometric recognition. With respect to templates generated using a CNN only, the discriminative capability of the created representations are improved by resorting to a cascade network composed of a backbone CNN and a sparse auto-encoder. The proposed system guarantees recognition performance better than the current state of the art on finger, palm, and dorsal vein patterns, with a GAR at FAR=10^{-3} equal to 99.99% on SDUMLA finger vein dataset [31], 99.69% on PolyU-P palm-vein dataset [32], and 87.45% on Bosphorus dorsum vein dataset [33]. The obtained results show that the sparseness in autoencoders has to be taken into account to improve the achievable error rates. In the performed tests, the SRIP loss, here applied for the first time to autoencoders, outperforms in most cases the commonly used D_{KL} and $L1$ losses.

REFERENCES

[1] B. Bhanu and A. Kumar, *Deep Learning for Biometrics*. Springer, 2017.

[2] K. Sundararajan and D. Woodard, "Deep Learning for Biometrics: A Survey," *ACM Computing Surveys*, vol. 51, no. 3, pp. 1–34, 2018.

[3] W. Yang, C. Hui, Z. Chen, J.-H. Xue, and Q. Liao, "FV-GAN: Finger Vein Representation Using Generative Adversarial Networks," *IEEE Transactions on Information Forensics and Security*, 2019.

[4] G. Wang, C. Sun, and A. Sowmya, "Multi-weighted Co-occurrence Descriptor Encoding for Vein Recognition," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 375–390, 2020.

[5] H. Qin, M. A. El-Yacoubi, J. Lin, and B. Liu, "An Iterative Deep Neural Network for Hand-vein Verification," *IEEE Access*, 2019.

[6] D. Thapar, G. Jaswal, A. Nigam, and V. Kanhangad, "PVSNet: Palm Vein Authentication Siamese Network Trained using Triplet Loss and Adaptive Hard Mining by Learning Enforced Domain Specific Features," in *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, pp. 1–8, IEEE, 2019.

[7] J. M. Song, W. Kim, and K. R. Park, "Finger-Vein Recognition Based on Deep DenseNet Using Composite Image," *IEEE Access*, 2019.

[8] J. Wang, Z. Pan, G. Wang, M. Li, and Y. Li, "Spatial Pyramid Pooling of Selective Convolutional Features for Vein Recognition," *IEEE Access*, vol. 6, pp. 28563–28572, 2018.

[9] Y. Fang, Q. Wu, and W. Kang, "A Novel Finger Vein Verification System Based on Two-Stream Convolutional Network Learning," *Neurocomputing*, vol. 290, pp. 100–107, 2018.

[10] E. Jalilian and A. Uhl, "Finger-Vein Recognition Using Deep Fully Convolutional Neural Semantic Segmentation Networks: The Impact of Training Data," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–8, IEEE, 2018.

[11] C. Xie and A. Kumar, "Finger Vein Identification using Convolutional Neural Network and Supervised Discrete Hashing," in *Deep Learning for Biometrics*, pp. 109–132, Springer, 2017.

[12] R. S. Kuzu, E. Maiorana, and P. Campisi, "Vein-based Biometric Verification using Transfer Learning," in *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pp. 403–409, IEEE, 2020.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-propagating Errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[15] J. V. Monaco, "Robust Keystroke Biometric Anomaly Detection," *arXiv preprint arXiv:1606.09075*, 2016.

[16] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Hoseini, and M. Fathy, "Online Signature Verification based on Feature Representation," in *2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pp. 211–216, IEEE, 2015.

[17] A. Sankaran, P. Pandey, M. Vatsa, and R. Singh, "On Latent Fingerprint Minutiae Extraction using Stacked Denoising Sparse Autoencoders," in *IEEE International Joint Conference on Biometrics*, pp. 1–7, IEEE, 2014.

[18] R. Raghavendra and C. Busch, "Learning Deeply Coupled Autoencoders for Smartphone based Robust Periocular Verification," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 325–329, IEEE, 2016.

[19] A. Majumdar, R. Singh, and M. Vatsa, "Face Verification via Class Sparsity Based Supervised Encoding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1273–1280, 2017.

[20] S. Gao, Y. Zhang, K. Jia, J. Lu, and Y. Zhang, "Single Sample Face Recognition via Learning Deep Supervised Autoencoders," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2108–2118, 2015.

[21] A. Kantarcı and H. K. Ekenel, "Thermal to Visible Face Recognition Using Deep Autoencoders," in *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pp. 1–5, 2019.

[22] M. Fayyaz, M. Hajizadeh-Saffar, M. Sabokrou, M. Hoseini, and M. Fathy, "A Novel Approach for Finger Vein Verification based on Self-taught Learning," in *2015 9th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pp. 88–91, IEEE, 2015.

[23] B. Hou and R. Yan, "Convolutional Auto-Encoder Based Deep Feature Learning for Finger-Vein Verification," in *IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2018.

[24] B. Hou and R. Yan, "Convolutional Autoencoder Model for Finger-Vein Verification," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 2067–2074, 2020.

[25] D. Thapar, G. Jaswal, A. Nigam, and V. Kanhangad, "PVSNet: Palm Vein Authentication Siamese Network Trained using Triplet Loss and Adaptive Hard Mining by Learning Enforced Domain Specific Features," in *International Conference on Identity, Security, and Behavior Analysis (ISBA)*, 2019.

[26] R. S. Kuzu, E. Maiorana, and P. Campisi, "Loss Functions for CNN-based Biometric Vein Recognition," in *28th European Signal Processing Conference (EUSIPCO 2020)*, 2020.

[27] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[28] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1492–1500, 2017.

[29] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive Angular Margin Loss for Deep Face Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.

[30] N. Bansal, X. Chen, and Z. Wang, "Can We gain more from orthogonality regularizations in training deep networks?," in *Advances in Neural Information Processing Systems*, pp. 4261–4271, 2018.

[31] Y. Yin, L. Liu, and X. Sun, "SDUMLA-HMT: A Multimodal Biometric Database," in *Chinese Conference on Biometric Recognition*, pp. 260–268, Springer, 2011.

[32] D. Zhang, Z. Guo, G. Lu, L. Zhang, and W. Zuo, "An Online System of Multispectral Palmprint Verification," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 2, pp. 480–490, 2009.

[33] A. Yüksel, L. Akarun, and B. Sankur, "Hand Vein Biometry based on Geometry and Appearance Methods," *IET Computer Vision*, vol. 5, no. 6, pp. 398–406, 2011.