Loss Functions for CNN-based Biometric Vein Recognition

Rıdvan Salih Kuzu*, Emanuele Maiorana†, and Patrizio Campisi‡
Section of Applied Electronics, Department of Engineering, Roma Tre University
Via Vito Volterra 62, 00146 Rome, Italy
*ridvansalih.kuzu@uniroma3.it,†emanuele.maiorana@uniroma3.it, ‡ patrizio.campisi@uniroma3.it

***Abstract*—The recent progress in deep learning has led to a rapid change in the way biometric data can be handled, offering new opportunities for further research on physical, behavioral, and cognitive biometric recognition. In particular, conventional modalities for preprocessing, extracting features, and comparing templates derived from biometric traits have been swiftly altered, replacing the search for hand-crafted features with the ever-increasing use of generalized deep learning models and transfer learning, able to guarantee notably-high recognition performance. This study investigates the capabilities of deep learning approaches in performing vein pattern verification. Specifically, recent advances in the design of convolutional neural networks, introduced to increase the inter-class variability and decrease the intra-class variability of the generated representations, are here taken into account to speculate on the effects on recognition performance of the selection for the most suitable loss function. Experimental tests conducted on finger vein, palm vein, and hand dorsum vein patterns testify the effectiveness of the proposed frameworks, able to exceed current state-of-the-art performance on five different publicly available vein datasets.**

***Index Terms*—Vein Biometrics, Finger Vein, Palm Vein, Hand Dorsum Vein, Convolutional Neural Networks, Loss Function.**

I. Introduction

Vein-based biometric systems [1] have been recently proven as reliable solutions for automatic people recognition, providing high recognition performance and notable robustness against digital impersonation [2]. With the rapid penetration of deep learning methods into the biometrics research field, it has also been noticed that significant improvements, with respect to current state of the art, can still be achieved before fully exploiting vein patterns discriminative capabilities [3].

In this paper, recent advances in the framework of convolutional neural networks (CNNs), specifically introduced to define feature representations characterized by a large inter-class variability and a small intra-class variability, are applied to vein patterns to design frameworks capable of guaranteeing verification rates exceeding current state of the art.

In more detail, networks defined for image classification tasks have been considered to extract distinctive features from the considered inputs, and properly modified to achieve better discriminability. Moreover, as an alternative to the standard softmax loss function commonly employed in most CNNs [4], additive angular margin penalty (AAMP) [5] and large margin cosine penalty (LMCP) [6] have been employed in the proposed networks. Eventually, in order to mitigate the lack of generalization in training due to the limited amount of data commonly available to train neural networks on publicly-available vein databases, we have exploited transfer learning to further boost the achievable recognition performance. To testify the effectiveness of the proposed approaches, they have been tested on three different vein modalities, namely finger vein, palm vein, and hand dorsum vein, considering five different public databases.

This paper is structured as follows: recent advances on vein-based biometric recognition are given in Section II, while the proposed vein verification system is outlined in Section III. Section IV details the considered benchmark datasets and the employed experimental frameworks, with the obtained results discussed in Section V, and conclusions drawn in Section VI.

II. State-of-the-Art: Vein Biometrics

Early efforts in performing vein-based biometric recognition have been described in [7], where Zhang et al. have extracted line features of palm vein and palm print, with the assistance of a texture-based coding algorithm and Gabor filters [7], achieving a 0.012% equal error rate (EER) on the multispectral PolyU-P dataset [7], yet comparing samples from the same session. Zhou and Kumar [8] have instead achieved 0.21% and 0.004% EERs on PolyU-P dataset with single-sample and six-sample enrolment data, respectively, and 4.51% and 1.37% EERs on CASIA dataset with single-sample and three-sample enrolment, employing Radon transform for template generation and modified Hamming distance for template comparison. Kumar et al. [9] have conducted additional tests on PolyU-F finger vein dataset, with a feature extraction scheme similar to the one proposed in [8], yet using an XOR-based distance for template comparison, achieving a 0.08% EER.

With the proliferation of deep learning techniques in recent years, the efforts towards the design of vein-based biometric recognition systems have also evolved. One of the early attempts at using convolutional neural networks (CNNs) on finger vein patterns has been accomplished by Radzi et al. [10], using a four-layered CNN followed by a convolutional-subsampling layer, and conducting experiments on an in-house dataset comprising data from 50 subjects. In another recent work by Fang et al. [11], two separate lightweight CNNs have been utilized for feature extraction, with a selective network automatically choosing one of them by evaluating intra-class and inter-class feature variations. For template comparison, support vector machines (SVMs) with linear kernels have been preferred. Notable recognition results have been achieved, with

The code of the study: https://github.com/ridvansalihkuzu/vein-biometrics

EERs at 0.47% and 0.10% on the SDUMLA and MMCBNU datasets, respectively, yet without properly separating training and testing pairs, and calculating EERs by taking the average of 5 test pairs. Generative adversarial networks (GANs) have also been exploited for finger vein biometric recognition in [12], where Yang et al. have achieved an EERs = 1.12% on the THU-FVFDT2 dataset, comprising finger vein samples acquired from 610 subjects during multiple sessions, and an EER = 0.94% on the SDUMLA dataset, in both cases splitting the available data into disjoint parts for training, validation and testing. The Densenet-161 CNN has been applied to extract distinctive characteristics from composite images, built by shifting the original finger vein samples towards different directions, in [13]. Experimental tests carried out on the PolyU-F and SDUMLA datasets have produced EERs at 0.33% and 2.35%, respectively, yet considering samples belonging to the same session for both enrolment and authentication purposes. Rather than using CNNs for feature extraction, Wang et al. have defined a multi-weighted descriptor encoding process, relying on convolutional filters, to extract co-occurrence features, compensating in this way the lack of diversity among samples from small-sized datasets [14]. To prove the effectiveness of their approach, they have performed experiments on finger, palm, and hand dorsum vein datasets. Several image segmentation CNN architectures have been instead used by Jalilian et al. [15] to extract binary image templates with the assistance of human-annotated pixel labels, reaching EERs at 0.64% and 2.63% on the UTFVP and SDUMLA finger vein datasets, respectively.

III. Proposed CNN-based Vein Recognition System

As mentioned in Section I, instead of defining novel CNN architectures from scratch to perform vein biometric recognition, we here exploit stable and consolidated networks, defined in literature for image classification tasks, to extract discriminative characteristics from the considered vein patterns. Specifically, preliminary tests in [16] have highlighted that Densenet-161 is the architecture able to guarantee the best recognition rates, when compared with Densenet-201 [17], VGG19 with batch normalization [18], and Inception-v3 [19]. We have therefore focused on Densenet-161, and modified its last layers to make it capable of generating more discriminative features from vein pattern inputs.

The proposed network, where a custom embedder layer has been added to the standard architecture, is described in Table I. In the reported configuration, each conv layer comprises the convolution, batch normalization, and ReLU stages. Furthermore, U in the classification layer refers to the total number of unique identities considered during training.

TABLE I
Proposed custom Densenet-161-based architecture.

Layers		**Input Size**	**Output Size**
Convolution	7×7 conv, str.2	$224 \times 224 \times 1$	$112 \times 112 \times 96$
Pooling	3×3 max pool, str.2	$112 \times 112 \times 96$	$56 \times 56 \times 96$
Dense Block 1	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$56 \times 56 \times 96$	$56 \times 56 \times 384$
Transition 1	1×1 conv	$56 \times 56 \times 384$	$28 \times 28 \times 192$
	2×2 avg pool, str.2		
Dense Block 2	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$28 \times 28 \times 192$	$28 \times 28 \times 768$
Transition 2	1×1 conv	$28 \times 28 \times 768$	$14 \times 14 \times 384$
	2×2 avg pool, str.2		
Dense Block 3	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$	$14 \times 14 \times 384$	$14 \times 14 \times 2112$
Transition 3	1×1 conv	$14 \times 14 \times 2112$	$7 \times 7 \times 1056$
	2×2 avg pool, str.2		
Dense Block 4	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$7 \times 7 \times 1056$	$7 \times 7 \times 2208$
Custom Embedder	7×7 global avg pool	$7 \times 7 \times 2208$	1×2208
	batch normalization		
	dropout (50%)		
	fully connected layer	1×2208	1×2048
	batch normalization		
Classifier	output layer	1×2048	$1\times U$

Recent advances in the design of CNNs for verification tasks have also been taken into account when selecting the loss functions to be employed for back-propagating the errors and estimating the network parameters. Specifically, in order to train the proposed network in standard identification modality, yet with the additional aim of achieving better feature embedding capabilities, i.e., generating representations having the largest possible inter-class variance as well as the smallest possible intra-class variance, loss functions such as the additive angular margin penalty (AAMP) [5] or the large margin cosine penalty (LMCP) [6], should be preferred over standard softmax.

To elaborate on, let $x_i \in \mathbb{R}^d$ denote the CNN-based feature representation of a sample belonging to the y_i-th class, and W_j denote the j-th column of the weight matrix $W \in \mathbb{R}^{d \times U}$ belonging to the last fully-connected layer of the considered CNN, where d and U are the feature size and the number of classes, respectively. Ignoring for simplicity the bias term when computing the logit, the activation of the final fully-connected layer (f_j) can be represented as:

$$f_j = W_j^\top x_i = ||W_j|| \cdot ||x_i|| \cdot \cos\theta_j, \tag{1}$$

being θ_j the angle between the weight W_j and the vein feature x_i. Thus, softmax loss function can be formulated as:

$$L_{softmax} = -\frac{1}{B}\sum_{i=1}^{B} log \frac{e^{f_{y_i}}}{\sum_{j=1}^{U} e^{f_j}} \tag{2}$$

where B is the batch size. Taking into account the L_2-normalized versions of both the features and the weights, that is, $\hat{x}_i$ and $\hat{W}_j$, the activation term in Equation (1) becomes $f_j = \cos\theta_j$, being $||\hat{x}_i|| = ||\hat{W}_j|| = 1$. The LMCP and AAMP loss functions can be then formulated as:

$$L_{LMCP} = -\frac{1}{B}\sum_{i=1}^{B} log \frac{e^{s(\cos(\theta_{y_i}) - m)}}{e^{s(\cos(\theta_{y_i}) - m)} + \sum_{j=1, j \neq y_i}^{U} e^{s \cos\theta_j}}, \tag{3}$$

$$L_{AAMP} = -\frac{1}{B}\sum_{i=1}^{B} log \frac{e^{s \cos(\theta_{y_i} + m)}}{e^{s \cos(\theta_{y_i} + m)} + \sum_{j=1, j \neq y_i}^{U} e^{s \cos\theta_j}}, \tag{4}$$

having set:

$$\hat{W}_j = \frac{W_j}{||W_j||}, \quad \hat{x}_i = \frac{x_i}{||x_i||}, \quad \hat{W}_j^\top \hat{x}_i = \cos\theta_j, \tag{5}$$

where the hyper-parameters s and m are the scale factor and the penalty margin, respectively. The functions in equations

TABLE II
DETAILS OF VEIN DATABASES USED IN THIS STUDY, AND EXPERIMENTAL PROTOCOLS OVER THEM.

Benchmark Database	Vein Modality	Database Statistics		Capturing Conditions	Capturing Parts	Summary of the Experimental Protocol
SDUMLA [20]	Finger	# of Subjects	106	grayscale single channel	index-, middle- and ring-fingers from left and right hand	In open-set scenario, classes are divided into two equal-size disjoint partitions; In closed-set scenario, the samples of each class are split into two equal parts for training and testing.
		# of Classes	636			
		# of Sessions	1			
		Samples per Session	6			
		Total Samples	3.816			
PolyU-F [9]	Finger	# of Subjects	156	grayscale single channel	index-middle-fingers from left hand	In open-set scenario, the classes with data collected in two sessions are split into two disjoint parts. In closed-set scenario, the first session is reserved for training, and the second one for testing.
		# of Classes	312			
		# of Sessions	2			
		Samples per Session	12			
		Total Samples	3.132			
PolyU-P [7]	Palm	# of Subjects	250	4 different spectral channels	left and right hand palm	In open-set scenario, the classes with data collected in two sessions are split into two disjoint partitions for training and testing, with enrolment and authentication samples taken from different sessions; In closed-set scenario, the first sessions are reserved for training, the second sessions for testing. Score fusion over channels is applied.
		# of Classes	500			
		# of Sessions	2			
		# per Session	6			
		Total Samples	24.000			
CASIA [21]	Palm	# of Subjects	100	6 different spectral channels	left and right hand palm	In open-set scenario, the classes with data collected in two sessions are split into two disjoint parts. In closed-set scenario, the first session is reserved for training, and the second one for testing. Only three channels ($700nm$, $850nm$, $940nm$) are considered.
		# of Classes	200			
		# of Sessions	2			
		Samples per Session	3			
		Total Samples	7.200			
Bosphorus [22]	Dorsum	# of Subjects	100	4 different environmental conditions	left hand dorsum	In open-set scenario, classes are divided into two equal-size disjoint partitions; In closed-set scenario, samples acquired in each environmental condition are split into two equal parts for training and testing. Test pairs for verification are chosen with samples in different conditions.
		# of Classes	100			
		# of Sessions	1			
		Samples per Session	3			
		Total Samples	1200			

(2), (3), and (4), can be employed to evaluate the loss for the network proposed in Table I, in order to estimate the parameters characterizing the sought vein representations.

During authentication, Euclidean distance is then used to evaluate the difference between the CNN-based representation derived from the query sample, and the one computed during user's enrolment. In case of multispectral data, available for instance in the PolyU-P and CASIA databases, each channel of the treated samples is fed to the corresponding network to obtain its feature representation as channel-wise 1024-dimensional vectors, with the scores from each channel then fused into a single value. The comparison of the computed output against a system threshold determines whether the presented subject is a legitimate user or an impostor.

A. Network Initialization and Optimization

As already mentioned, the considered networks are trained in identification mode, with the outputs of the proposed custom embedder layer taken during testing as feature templates for verification purposes. During training, stochastic gradient descent (SGD) with a batch size of 64, a learning rate of 0.01 divided by 10 after every 30 epochs, a momentum of 0.9 for speeding up the convergence of gradient vectors, and a maximum number of 120 training epoch, is employed. As for the hyper-parameters of AAMP and LMCP loss functions, the penalty margin is selected in the range $m \in [0.3 - 0.7]$ with a step size of 0.05, and the scale parameter is selected in the range $s \in [16 - 96]$ with a step size of 16.

Default initialization parameters of PyTorch Library are initially used for Densenet-161. Then, to exploit the effectiveness of transfer learning, also the weights of the network pre-trained on Imagenet are employed for initialization. Such an approach has been tested to evaluate whether the issue of having limited datasets for network training can be mitigated by exploiting stable models, already trained on large-scale databases. When employing Imagenet initialization for the modified Densenet-161, the proposed custom layers are initialized using unit weight initialization for batch normalization, and Glorot uniform initialization for fully-connected layers.

TABLE III
EERS ON ORIGINAL AND MODIFIED DENSENET-161 ARCHITECTURES.

Network	Database				
	SDUMLA	PolyU-F	PolyU-P	CASIA	Bosphorus
Densenet-161	1.73%	12.40%	2.52%	22.30%	8.92%
Modified Densenet-161	1.45%	7.43%	0.14%	13.80%	7.00%

IV. EXPERIMENTAL FRAMEWORK

The proposed approach for vein-based biometric recognition has been tested on five publicly available vein databases, comprising samples of finger vein, palm vein, and hand dorsum vein patterns. Specifically, we have performed tests on the PolyU-F finger vein [9] and the PolyU-P multispectral palm [7] databases of the Hong Kong Polytechnic University, on the SDUMLA [20] finger vein database of the Shandong University, on the CASIA multispectral palm vein database [21] of the Chinese Academy of Sciences' Institute of Automation, and on the Bosphorus hand dorsum vein database [22] of the Boğaziçi University.

As summarized in Table II, different fingers and different hands of the users are treated as different classes. For each dataset, after dividing it into two equally-sized parts for training and testing, 20% of the training data is reserved for validation. Open-set scenarios, in which data from a given class are not used in both training and testing, have been considered. For multi-session databases, pairs for enrolment and authentication are chosen from different sessions.

All samples have been resized to 224×224 pixels, and normalized to zero mean and unit variance, before feeding them into the proposed network. PyTorch 1.1.0 with a system

TABLE IV

EERs REACHED ON EACH DATASET FOR DIFFERENT LOSS FUNCTIONS AND DIFFERENT NETWORK INITIALIZATIONS.

Database	Default Initialization						ImageNet Initialization					
	Open-set Scenario			Closed-set Scenario			Open-set Scenario			Closed-set Scenario		
	Softmax	LMCP	AAMP	Softmax	LMCP	AAMP	Softmax	LMCP	AAMP	Softmax	LMCP	AAMP
SDUMLA	1.45%	0.68%	0.87%	0.27%	0.04%	0.14%	0.40%	0.08%	0.02%	0.04%	0.01%	0.01%
PolyU-F	7.43%	4.48%	3.69%	1.21%	0.86%	1.01%	5.57%	2.17%	1.87%	0.13%	0.00%	0.00%
PolyU-P	0.14%	0.03%	0.04%	0.04%	0.02%	0.02%	0.01%	0.00%	0.00%	0.00%	0.00%	0.00%
CASIA	13.80%	9.73%	8.79%	3.45%	3.36%	2.87%	2.26%	1.43%	1.12%	1.15%	0.42%	0.46%
Bosphorus	7.00%	6.36%	4.78%	2.30%	1.03%	1.26%	5.63%	3.10%	2.58%	0.70%	0.23%	0.00%

configuration of 32Gb RAM, two NVIDIA$^{\text{TM}}$ Titan V graphic cards, i7-3.4GHz processor, and Windows$^{\text{TM}}$ 10 OS, are used.

V. Results and Discussion

As shown in Table III, where results obtained in open-set scenarios with softmax loss are reported, the proposed modified Densenet-161 performs notably better than the standard version for vein-based biometric recognition. Nonetheless, poor results are still reached on PolyU-F, CASIA, and Bosphorus datasets.

In order to improve the achievable performance, LMCP and AAMP losses have to be employed together with the custom Densenet-161 CNN here introduced. As shown in Table IV, both functions allows EERs significantly better than those achieved with softmax.

Table IV also shows that further improvements can be achieved exploiting transfer learning, as described in Section III-A: initializing the modified Densent-161 with weights pre-trained on Imagenet, EERs outperforming state-of-the-art results on all the considered databases have been achieved using AAMP as loss function.

In more detail, we have reached 0.02% EER on open-set scenario over the SDUMLA dataset, when the best EERs reported in literature are 0.94% by Yang et al. [12] for open-set scenario, and 0.47% by Fang et al. [11] for closed-set scenario.

As for the PolyU-F dataset, when False Acceptance Rate (FAR) is 0.01%, the usage of AAMP in training the model with default initialization improves the Genuine Acceptance Rates (GAR) of about 18%. Moreover, initializing the network with pre-trained weights improves drastically GAR from 24% to 78%. The obtained EER of 1.87% exceeds any other study in the literature, with the only exception of the study in [13], reporting a 0.33% EER. However, only the first session of the PolyU-F dataset has been there used to form enrolment and authentication pairs. When following the same experimental approach in [13], the proposed method is able to reach 0.16% EER, which is about 50% better than the results in [13].

On the PolyU-P dataset, Wang et al. [23] and Qin et al. [24] have respectively achieved 0.017% and 0.015% EERs in closed-set conditions, while Zhang et al. [7] have achieved 0.012% EER by pairing same-session images. We have instead reached 0% EER for both open-set and closed-set training/test partitions using AAMP and Imagenet initialization on the modified Densenet-161 model.

When considering the multispectral CASIA dataset, separately exploiting the contributions at each wavelength has led to EERs of 2.57%, 2.42%, and 2.10%, respectively for samples taken at 700nm, 850nm, 930nm illumination. However, when the outputs of the three separate CNN models are jointly used, an EER at 1.12% has been achieved employing AAMP and Imagenet weight initialization. The proposed approach therefore outperforms the methods in [8] and [25]. It has to be mentioned that an EER at 0.33% has been reported in [24], and that Wu et al. [26] have also achieved an EER at 0.029%. Yet, the network used in [24] has been trained on the same subjects then used for testing purposes, while no details about how samples have been arranged into training, enrolment, and authentication subsets, are given in [26].

Finally, tests on the Bosphorus dataset have resulted in an EER = 2.58% with single-sample enrolment in the proposed approach. Furthermore, we have reached 0% EER in closed-set verification scenario. The recognition rates reported by Yüksel et al. [22] instead consist of a 8.18% EER for single-sample enrolment, and a 2.25% EER for eight-sample enrolment.

Figure 1 reports the receiver operating characteristics (ROC) of the performance achieved in open-set scenarios on the five considered datasets with the proposed approaches, as well as the genuine and impostor score distributions generated when training the proposed modified Densenet-161 model with AAMP loss function and Imagenet initialization.

VI. Conclusion

In this paper, we have proposed a novel vein-based recognition system using CNNs. By modifying the Densenet-161 architecture, optimizing it with additive angular margin penalty, and making use of the weights pre-trained on Imagenet dataset, we have outperformed the state-of-the-art recognition performance on some of the benchmark vein datasets. Specifically, considering open-set scenarios, and taking enrolment and authentication pairs from different sessions whenever possible, we have reached EER = 0.02% on SDUMLA finger vein dataset [20], EER = 1.87% on PolyU finger vein dataset [9], EER = 0% on PolyU palm vein dataset [7], EER = 1.12% on CASIA multispectral palm dataset [21], and EER = 2.58% on Bosphorus hand dorsum vein dataset [22].

The obtained results show that loss functions forcing an increased inter-class variability and a decreased intra-class variability for the generated representations allow achieving high-level verification results on all the considered kinds of vein patterns. Moreover, initializing networks with weights pre-trained of large datasets further boosts vein-based biometric recognition performance.

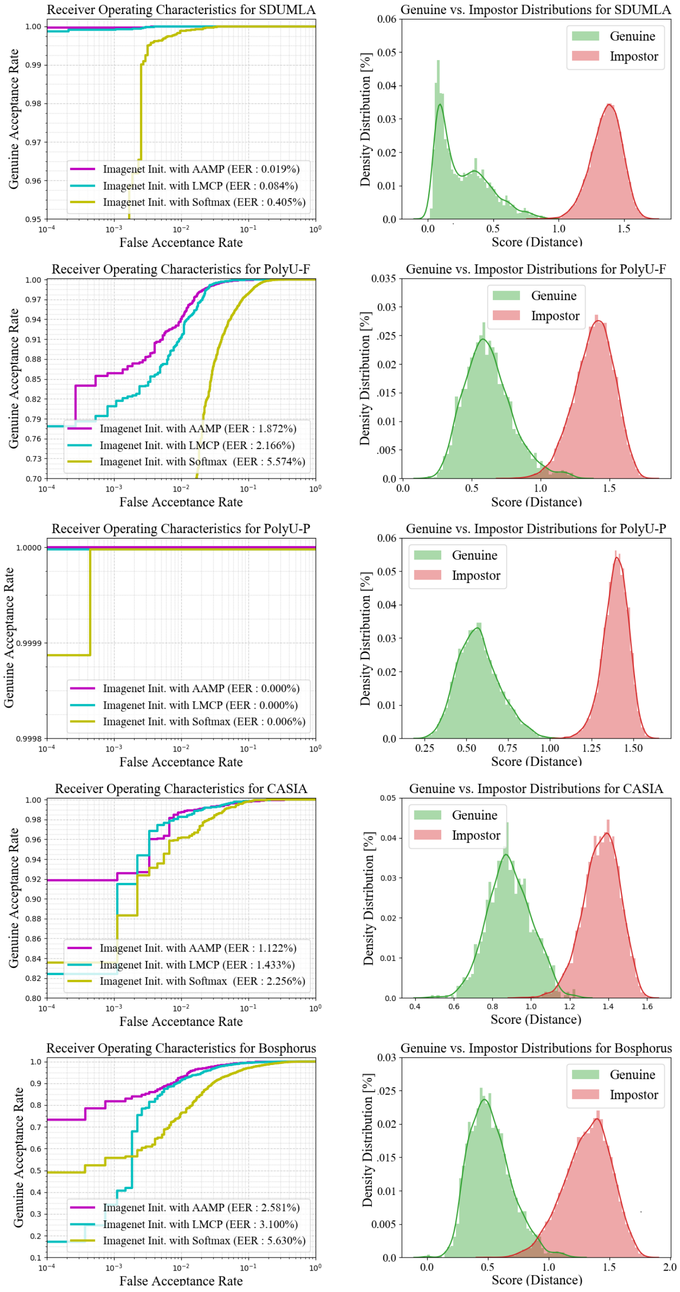

Fig. 1. Performance in open-set scenarios: ROCs for different loss functions (left), and score distributions for AAMP and Imagenet initialization (right).

References

[1] Naoto Miura, Akio Nagasaka, and Takafumi Miyatake, "Extraction of Finger-Vein Patterns Using Maximum Curvature Points in Image Profiles," *IEICE TTransactions on Information and Systems*, vol. 90, no. 8, pp. 1185–1194, 2007.

[2] E. Piciucco, E. Maiorana, C. Kauba, A. Uhl, and P. Campisi, "Cancelable biometrics for finger vein recognition," in *Int. Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*, 2016.

[3] R. Das, E. Piciucco, E. Maiorana, and P. Campisi, "Convolutional neural network for finger-vein-based biometric identification," *IEEE Trans. on Information Forensics and Security*, vol. 14, no. 2, pp. 360–373, 2019.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, Adaptive Computation and Machine Learning Series. MIT Press, 2017, Chapter 6.5: Gradient-Based Learning, pp. 178-182.

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

[6] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu, "Cosface: Large Margin Cosine Loss for Deep Face Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[7] David Zhang, Zhenhua Guo, Guangming Lu, Lei Zhang, and Wangmeng Zuo, "An Online System of Multispectral Palmprint Verification," *IEEE Trans. on Instr. and Measurement*, vol. 59, no. 2, pp. 480–490, 2009.

[8] Yingbo Zhou and Ajay Kumar, "Human Identification Using Palm-Vein Images," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, pp. 1259–1274, 2011.

[9] Ajay Kumar and Yingbo Zhou, "Human Identification Using Finger Images," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2228–2244, 2011.

[10] Syafeeza Ahmad Radzi, Mohamed Khalil Hani, and Rabia Bakhteri, "Finger-Vein Biometric Identification using Convolutional Neural Network," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 24, no. 3, pp. 1863–1878, 2016.

[11] Yuxun Fang, Qiuxia Wu, and Wenxiong Kang, "A Novel Finger Vein Verification System based on Two-stream Convolutional Network Learning," *Neurocomputing*, vol. 290, pp. 100–107, 2018.

[12] Wenming Yang, Changqing Hui, Zhiquan Chen, Jing-Hao Xue, and Qingmin Liao, "FV-GAN: Finger Vein Representation Using Generative Adversarial Networks," *IEEE Trans. on Information Forensics and Security*, 2019.

[13] Jong Min Song, Wan Kim, and Kang Ryoung Park, "Finger-Vein Recognition Based on Deep DenseNet Using Composite Image," *IEEE Access*, 2019.

[14] 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.

[15] Ehsaneddin Jalilian and Andreas 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)*. IEEE, 2018, pp. 1–8.

[16] R.S. Kuzu, E. Maiorana, and P. Campisi, "Vein-based biometric verification using transfer learning," in *International Conference on Telecommunications and Signal Processing (TSP)*, 2020.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[18] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition," in *International Conference on Learning Representations*, 2015.

[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[20] Yilong Yin, Lili Liu, and Xiwei Sun, "SDUMLA-HMT: A Multimodal Biometric Database," in *Chinese Conference on Biometric Recognition*. Springer, 2011, pp. 260–268.

[21] Ying Hao, Zhenan Sun, Tieniu Tan, and Chao Ren, "Multispectral Palm Image Fusion for Accurate Contact-Free Palmprint Recognition," in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 281–284.

[22] Aycan Yüksel, Lale Akarun, and Bülent Sankur, "Hand Vein Biometry based on Geometry and Appearance Methods," *IET Computer Vision*, vol. 5, no. 6, pp. 398–406, 2011.

[23] Jun Wang, Zaiyu Pan, Guoqing Wang, Ming Li, and Yulian Li, "Spatial Pyramid Pooling of Selective Convolutional Features for Vein Recognition," *IEEE Access*, vol. 6, pp. 28563–28572, 2018.

[24] Huafeng Qin, Mounim A El-Yacoubi, J. Lin, and Bo Liu, "An Iterative Deep Neural Network for Hand-vein Verification," *IEEE Access*, 2019.

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

[26] W. Wu, S. J. Elliott, S. Lin, and W. Yuan, "Low-cost Biometric Recognition System based on NIR Palm Vein Image," *IET Biometrics*, vol. 8, no. 3, pp. 206–214, 2019.