SigmML: Metric meta-learning for Writer Independent Offline Signature Verification in the Space of SPD Matrices

Alexios Giazitzis and Elias N. Zois
Telsip Laboratory, UniWA
Petrou Ralli & Thivon Ave., 12241, Egaleo, Greece
{ene262017050, ezois}@uniwa.gr

Abstract

The handwritten signature has been identified as one of the most popular biometric means of human consent and/or presence for transactions held by any number of physical or legal entities. Automated signature verification (ASV), merge popular scientific branches such as computer vision, pattern recognition and/or data-driven machine learning algorithms. Up to now, several metric learning approaches for designing a writer-independent signature verifier, have been developed within a Euclidean framework by means of having their operations closed with respect to real vector spaces. In this work, we propose, for the first time in the ASV literature, the use of a meta-learning framework in the space of the Symmetric Positive Definite (SPD) manifold as a means to learn a pairwise similarity metric for writer-independent ASV. To begin, pairs of handwritten signatures are converted into a multidimensional distance vector with elements corresponding SPD distances between spatial segments of corresponding covariance pairs. We propose a novel meta-learning approach which explores the structure of the input gradients of the SPD manifold by means of a recurrent model, constrained by the geometry of the SPD manifold. The experimental protocols utilize two popular signature datasets of Western and Asian origin in two blind-intra and blind-inter (or cross-lingual) transfer learning approach. It also provide evidence of the discriminating nature of the proposed framework at least when summarized against other State-of-the-Art models, typically realized under a framework of Euclidean, or vector space, nature.

1. Introduction

Signature verification (SV), the authentication or consent of human actions by means of the handwritten signature, is a captivating biometric task [48], usually performed by a Forensic Handwriting Examiner. Today, automated SV is an active research field, since it incorporates a unique kind of behavioral data, derived by a combined product of a neuromotor skill and a taughting system [2]. Their natural variability poses an important element for competent human verification [46], by merging diverse scientific fields which models the resulted cognitive task usually depicted by an optical pen trace [13].

Surveys conducted on SV systems up until now [20, 36, 48], may categorize the systems as offline/static and online/dynamic, according to the acquisition mode, *i.e*. digital images or time-indexed feature sequences. Another categorization that exists for SV systems depends on the approach towards the verifier [9,30,45,47,53]. A system that operates on a per-writer basis, is termed Writer-Dependent (WD), while a system that learns to discriminate similarities, or not, between signature pairs is termed Writer-Independent (WI). Thus, model differences are formed, usually in the form of the dichotomy transform (DT) [41], where a feature space $F \subseteq \mathbb{R}^d$ that contains any two pairs of signature representations (s_i, s_j), is mapped to a distance space $D = |s_i - s_j| \subseteq \mathbb{R}^d_+$, termed (dis)similarity space, that contains the element-wise distances of any two feature vectors s_i and s_j. This, and transformations of similar nature, creates two distributions: a) the ω^+ or positive distribution, composed by the genuine-to-genuine pairs, and b) the ω^- or negative distribution, encompassing the genuine-to-forgery pairs.

An important concern when a binary WI-SV verifier is designed addresses the way that the negative distribution ω^- is formed, given the fact that there are numerous types of forgeries. A literature search [34–36, 44, 45, 53], shows that forgery samples may be assigned as a) *Random* or b) *Simple* if the forgery level is simplistic and c) *Simulation*: or d) *Skilled* if the forger mimics the genuine style of handwriting. Further, an unbiased assessment of any given WI-SV verifier, requires that the signature pairs used for the development stage should not be participants in the testing stage.

In this paper, we formulate the SV problem as a metric learning problem in the space of SPD matrices by means

of the Mahalanobis distance, for which we develop a meta-learning framework in order to learn the SPD matrix parameter required by the distance formula. To that end we propose:

1. A novel mapping for pairs of SPD matrices into a distance vector, termed hereafter as the SPD Distance Pyramid (DP). The intuition behind DP is that, contrary to measuring a single distance between two SPD matrices, we map the original pairs of signature covariance matrices to a distance vector with elements an assortment of SPD related distances. One may contemplate an analogy between the typical DT, that maps pairs of features vectors to the (dis)similarity space. In our case, both the representation as well as the mapping to the distance space have a manifold nature; images are coded into SPD matrices while pairs of them are mapped into a distance vector with elements SPD measures.

2. A novel offline WI-SV metric Meta-Learning algorithm, called SigmML, which utilises the meta-learning network of [17], with the Mahalanobis distance. The objective is to calculate a covariance matrix which will minimise the distance of similar signatures while maximize dissimilar ones. We modify the learning algorithm of the network to jointly train it with the covariance parameter. As the task at hand utilizes data that are not easy to acquire, we take advantage of them with meta-learning, which exploits underlying information of the gradients produced from the SPD parameter. Although the SPD manifold has been used in the formulation of an SV system in [54], it is WD, without meta-learning or as a metric learning task.

3. The use of three ω^{-} assemblies in order to evaluate the efficiency of the proposed WI-SV architecture. We realize the following ω^{-} implementations: a) only genuine-to-skilled forgeries, b) genuine-to-random and genuine-to-skilled forgeries in equal amount and, c) only genuine-to-skilled forgery pairs. We utilise disjoint training, validation and testing sets. Additionally, the use of blind learning and testing sets allows us to proceed with the use of genuine-to-skilled forgery pairs without any bias induce.

Our source code is available at a https://bitbucket.org/agiaz/sigmml for reproducibility purposes.

Section 2 summarize the literature for metric learning based WI-SV systems. Section 3 provides an overview of the proposed architecture and the steps needed for the formation of DP vector. Section 4 reviews the elements of the SPD Riemannian manifold. Section 5 provides all the necessary details regarding the proposed SPD metric meta-learning framework. Section 6 describes the experimental protocol and provides the results. Finally, section 7 provides the conclusion.

2. Metric learning in SV

Metric learning has been noted to be among the main approaches for SV systems. It is easy to understand why; as a principle in SV, learning a metric which separates similar from dissimilar objects [18] is approachable, easy to understand and can be formulated with seemingly any kind of algorithmic basis. As seen in [20], metric learning based algorithms are performed with the assumption that the appropriate representation to perform the SV task belongs to a vector space, regardless of whether the vector representation of a signature image is either learned or handcrafted. Although this assumption has produced a number of State-of-the-Art (SOTA) algorithms, as seen in Table 1, it is not necessarily the best approach. As seen in [54], even though it is not a metric learning approach and it is a WD framework, the SPD representation of the signature images can provide equally efficient results.

In Table 1, we provide a concise summary of the SOTA algorithms for the SV task formulated under a metric learning framework, the features used to generate the vector representation and the datasets utilized for training and testing. It is easy to observe that DCNN representation learning is now the dominant approach, since the deep learning nature could identify and extract features for which there may be no handcrafted algorithm yet. Most of the presented methods were trained and tested on multiple datasets, which provides a measure of their ability to discriminate between genuine and forgery signatures, internally on the dataset they were trained and externally on other datasets that were used.

3. Overview of the Proposed Architecture

Figure 1 depicts a graphical overview to the proposed WI-SV framework. To begin, any signature image I_{raw} is first subjected to a typical preprocessing stage, described in the literature [55]. In summary, it is comprised from the following steps: binarization, using Otsu's method, followed by a thinning operation. The resulted thinned binary image is the mask operator which extracts the relevant gray-scale information I from the original image I_{raw}. Next, a total of ten image filters described by Eq. (1) is applied in I in order to create a corresponding ten layered image stack, $F(I)$.

$$F(I) = \left\{ I, |I_x|, |I_y|, \sqrt{I_x^2 + I_y^2}, \tan^{-1}\left(\frac{I_y}{I_x}\right), \right. \\ \left. |I_{xx}|, |I_{yy}|, |I_{xy}|, x, y \right\} \quad (1)$$

Each layer contains the grayscale intensity I, the directional first and second order derivatives I_x and I_y,

1st Author #Ref	Method/Features	Datasets
Maergner [33]	Multiple classifier system (MCS) / Graph Edit Distance (GED) + DCNN features	GPDSsynthetic [14], MCYT-75 [38], UTSig [44], CEDAR [24]
Maergner [32]	MCS / GED + Inkball models	GPDSsynthetic, MCYT-75
Soleimani [43]	Deep Multitask Metric Learning (DMML) / Histogram of Oriented Gradients (HOG), Discrete Radon Transform (DRT)	UTSig, MCYT-75, GPDSsynthetic, GPDS960GraySignatures [15]
Liu [28]	Deep Convolutional Siamese Network (DCSM) / Mutual Signature DenseNet (MSDN)	CEDAR, GPDSsynthetic, ChnSig [28]
Zhu [52]	Point-to-Set (P2S) Distance / DCNN	ChiSig, CEDAR, MCYT-75, BiosecurID-SONOF [16], SigComp11 [29]
Hamadene [19]	Feature Dissimilarity Measure Thresholding (FDMT) / Contourlet Transform (CT)	CEDAR. GPDS300 [8]
Parcham [40]	CBCapsNet / CBCapsNet	CEDAR, GPDS300, GPDSsynthetic, BHsig260 [39]
Hanif [21]	Mahalanobis Distance / HOG + Local Binary Patterns (LBP) + Vector of Locally Aggregated Descriptors (VLAD)	CEDAR, BHsig260
Chattopadhyay [12]	Self-Supervised pre-trained + transfer learned ResNet-18/ SURDS	BHsig260
Lin [27]	Euclidean Distance / 2-Channel-2-Logit (2C2L)	GPDSsynthetic, CEDAR, BHsig260
Lu [31]	Adaptive Distance Fusion (ADF) / Spatial Transformer Network + Attentive Recurrent Comparator	CEDAR, GPDSsynthetic, BHsig-260
Natarajan [37]	Euclidean distance / SigNet based Siamese Network	SigComp11, CEDAR, BHsig260
Viana [49]	Euclidean distance / Multi-task handwritten signature representation learning (MHSRL)	GPDSsynthetic, CEDAR, MCYT-75
Wan [50]	Euclidean distance / SigCNN	GPDSsynthetic, MCYT-75, CEDAR, CSIG-WHU [50]
Ren [42]	Euclidean Distance / Signature Graph Convolutional Network (SigGCN)	CEDAR, BHsig-260
Huang [23]	Eucldiean Distance / Multiscale Global and Regional Feature Learning Network (MGRNet)	CEDAR, BHsig260, HanSig [23]

Table 1. Summary of the State-of-the-Art metric learning SV systems.

I_{xx}, I_{yy} and I_{xy}, the magnitude and direction of the gradient and the normalized row x and column y for each signature trace pixel. It is noted that the evaluation of the signature image covariance matrix is only with respect to the pixels of the signature trace I provided by the preprocessing stage. The image covariance matrix $\mathbf{C}$ is then computed (see supplementary). Hereafter, capital bold letters, *e.g.*, $\mathbf{C}$, denote SPD matrices, while capital italic letters, *e.g.*, X, denote symmetric matrices.

The Distance Pyramid. Let us recall the Rao's Distance, in Eq. (2), as the distance between two SPD points $(\mathbf{C}^i, \mathbf{C}^j)$ induced by equipping the SPD Manifold with the the Rao-Fisher metric [4].

$$d(\mathbf{C}^i, \mathbf{C}^j) = \left\| \text{logm}((\mathbf{C}^i)^{-\frac{1}{2}} \mathbf{C}^j (\mathbf{C}^i)^{-\frac{1}{2}}) \right\|_F \qquad (2)$$

where, $\| \cdot \|_F$ is the Frobenius norm, and logm denotes the matrix logarithm, computed by applying the element-wise log function to the eigenvalues given by the spectral decomposition of any real symmetric matrix X: $\text{logm}(X) = \text{U}\log(D)\text{U}^T$.

Given a pair of covariance matrices $(\mathbf{C}^i, \mathbf{C}^j)$, we can now define their similarity vector representation, termed as the Distance Pyramid (DP) vector, to be a set of real numbers with elements the union of local distances $\text{DP}_{\{a,b\}} = \bigcup\{d(\mathbf{C}^i, \mathbf{C}^k)_{a:b})\}$ between corresponding (sub)SPD block

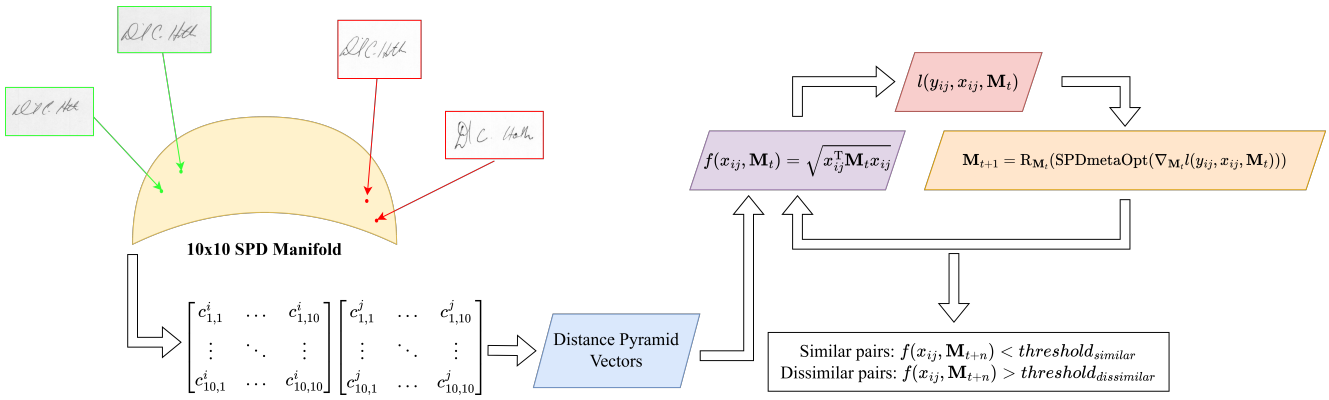

Figure 1. Block diagram overview of the proposed meta learning framework. For any two signatures (C^i, C^j) that are mapped to $\mathcal{S}_{10}^{+}$, a pairwise distance vector is computed based on the size of the windows and if they will be overlapping.

diagonal matrices $\mathbf{C}^i_{a:b}, \mathbf{C}^j_{a:b}$ of the primary $(\mathbf{C}^i, \mathbf{C}^j)$ covariance pair. An element $d(\mathbf{C}^i, \mathbf{C}^k)_{a:b})$ of the $\mathrm{DP}_{\{a,b\}}$ is evaluated using Eq. (2) between two (sub)SPD pairs $(\mathbf{C}^i_{a:b}, \mathbf{C}^j_{a:b})$ in which $a : b$ points to the (sub)SPD, block diagonal matrix of $\mathbf{C}^i$ with indices ranging from a to b. Thus, setting $\{a, b\}$ to a number of configurations unfolds the DP structure to several dimensionalities. For example, the $\mathrm{DP}_{\{1,10\}} = \{d(\mathbf{C}^i, \mathbf{C}^k)_{1:10})\} = d(\mathbf{C}^i_{1:10}, \mathbf{C}^j_{1:10})$ asserts that the DP structure is a single real valued number, while the $\mathrm{DP}_{\{1,10\},\{1,3\},\{2,7\}} = \{d(\mathbf{C}^i, \mathbf{C}^k)_{1:10}), d(\mathbf{C}^i, \mathbf{C}^k)_{1:3}), d(\mathbf{C}^i, \mathbf{C}^k)_{2:7})\}$ states that the DP structure is comprised from three real valued numbers. For the purpose of this submission, we propose to explore two distinct DP structures by utilizing a moving window, that can either overlap with itself or not, and selects (sub)SPD block diagonal matrices. The first one, termed as non-overlapping $\mathrm{DP}^{NOL} = \mathrm{DP}_{\{1,10\},\{1,5\},\{6,10\}} = \{d(\mathbf{C}^i, \mathbf{C}^k)_{1:10}), d(\mathbf{C}^i, \mathbf{C}^k)_{1:5}), d(\mathbf{C}^i, \mathbf{C}^k)_{6:10})\}$ maps the $(\mathbf{C}^i, \mathbf{C}^j)$ covariance pair into a three dimensional vector, while the second one, termed as overlapping $\mathrm{DP}^{OL} = \mathrm{DP}_{\{1,10\},\{1,5\},\{2,6\},\ldots,\{6,10\},\{1,7\},\ldots,\{4,10\}}$ maps the $(\mathbf{C}^i, \mathbf{C}^j)$ covariance pair into a resultant eleven dimensional vector. In Figure 2, we provide a toy example of the process of creating two different DP vectors with a window of size 5×5 in both overlapping and non overlapping modes of operation.

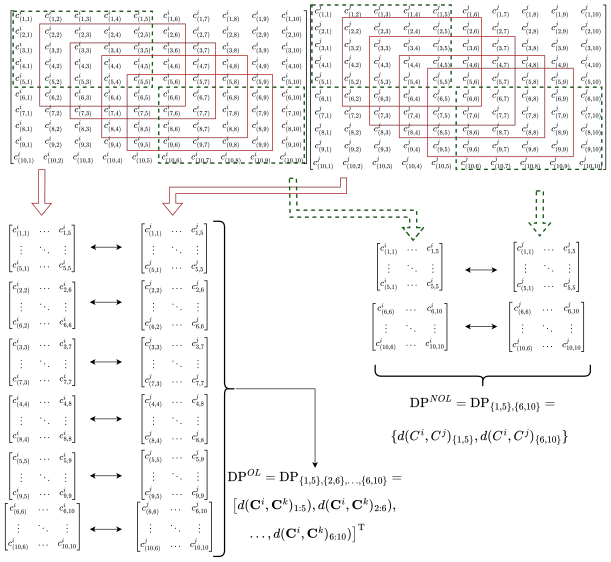

Figure 2. Toy examples of two DP vectors: Both red and dashed green windows are used for the creation process of a DP overlapping vector $\mathrm{DP}^{OL} = \mathrm{DP}_{\{1,5\},\{2,6\},\ldots,\{6,10\}}$ with the use of a 5×5 overlapping window. Only the green dashed windows are used in the creation process of a DP non-overlapping vector $\mathrm{DP}^{NOL} = \mathrm{DP}_{\{1,5\},\{6,10\}}$ with the use of a 5×5 non overlapping window.

4. Elements of the SPD Manifold

The space of all d-dimensional SPD matrices $\mathcal{S}_d^{+}$ is a Riemannian submanifold [1, 10], as it is a non-vector space of d^2 dimensions, from which only $^{d(d+1)}/_{2}$ are independent. The geometry of the manifold is frequently exploited by equipping it with the Rao-Fisher metric [4], which allows us to perform operations between two SPD operands $\mathbf{Y}$, $\mathbf{Z}$, similar to the basic operations of addition and subtraction of the Euclidean space.

Points on the SPD manifold are mapped to the tangent space of another manifold point, which is a vector space with an inner product, to perform operations. The mapping of an SPD point $\mathbf{Y}$, to the tangent plane $\mathrm{T}_{\mathbf{X}}\mathcal{S}_d^{+}$, the vector space of symmetric matrices, of a SPD point $\mathbf{X}$, is computed with the logarithmic map of Eq. (3) while the inverse, the mapping of a tangent vector Y to the SPD manifold, use the exponential map, by Eq. (4).

$$Y \equiv \log_{\mathbf{X}}(\mathbf{Y}) = \mathbf{X}^{\frac{1}{2}} \mathrm{logm}\left(\mathbf{X}^{-\frac{1}{2}} \mathbf{Y} \mathbf{X}^{-\frac{1}{2}}\right) \mathbf{X}^{\frac{1}{2}} \qquad (3)$$

$$\mathbf{Y} \equiv \exp_{\mathbf{X}}(Y) = \mathbf{X}^{\frac{1}{2}} \operatorname{expm}\left(\mathbf{X}^{-\frac{1}{2}} \mathbf{Y} \mathbf{X}^{-\frac{1}{2}}\right) \mathbf{X}^{\frac{1}{2}} \quad (4)$$

for which expm is calculated like logm, *i.e.*, $\operatorname{expm}(X) = U \exp(D) U^{\mathrm{T}}$, where exp denotes the element-wise exponential function. The Rao-Fisher metric induces the geodesic distance between two SPD points, and its is the Rao's distance, formerly presented in Eq. (2).

An important concept in the optimization perspective of machine learning is gradient descent, with which we optimize a formulated objective function, $l(x, y, w_{(t)})$, for the task we wish to solve. Gradient descent basically utilizes the gradients, $\nabla_{w_{(t)}} l$, generated by the differentiation of an objective function w.r.t. some learnable parameters, $w_{(t)}$ in order to update them in a way that optimizes the objective function. For example, in the Euclidean space, updating $w_{(t)}$ to $w_{(t+1)}$ is performed by adding a fraction (λ - learning rate) of the gradient, $\nabla_{w_{(t)}} l$, to the previous learnable parameter, *i.e.*, $w_{(t)}$, $w_{(t+1)} = w_{(t)} + \lambda \nabla_{w_{(t)}} l(x, y, w_{(t)})$, where x, y are the input and output data. When the learnable parameter belongs to the SPD manifold. *e.g.*, $\mathbf{W}_{(t)}$, gradient descent needs to utilize its' tangent space, $\mathrm{T}_{\mathbf{W}_{(t)}} \mathcal{S}_d^+$. In SPD manifolds, the gradient descent update step is performed with the use of the retraction operation [1, 10], in which a $\mathbf{W}_{(t)}$ can be updated towards the direction pointed by its' gradient, $\nabla_{\mathbf{W}_{(t)}}$, which actually resides in its' tangent space. The SPD retraction operation is given by Eq. (5), which corresponds to the exponential map of the SPD manifold. Usually, in popular deep learning libraries, an automatic differentiation process is implemented [5, 11], which calculates gradients without caring about any kind of preserving the structure of the learnable parameters, resulting in gradients that are not necessarily part of the tangent space of any Riemannian manifold point. A workaround, without the need to reimplement any part of the library, is to use a projection operation, $\pi_{\mathbf{M}}(\cdot)$, that takes as input a Euclidean vector, X, and projects it to the tangent plane of a point $\mathbf{M}$. For the SPD manifold, the projection operation is given by Eq. (6). This leads to the frequent use of Eq. (7).

$$\mathrm{R}_{\mathbf{W}_{(t)}}\left(\nabla_{\mathbf{W}_{(t)}}\right) = \exp_{\mathbf{W}_{(t)}}\left(-\lambda \nabla_{\mathbf{W}_{(t)}}\right) \quad (5)$$

$$\nabla_{\mathbf{W}_{(t)}} \equiv \pi_{\mathbf{W}_{(t)}}\left(\nabla_{w_{(t)}}\right) = \mathbf{W}_{(t)} \left(\frac{\nabla_{w_{(t)}} + \nabla_{w_{(t)}}^{\mathrm{T}}}{2}\right) \mathbf{W}_{(t)} \quad (6)$$

$$\mathrm{R}_{\mathbf{W}_{(t)}}\left(\nabla_{w_{(t)}}\right) = \exp_{\mathbf{W}_{(t)}}\left(-\lambda \pi_{\mathbf{W}_{(t)}}\left(\nabla_{w_{(t)}}\right)\right) \quad (7)$$

Gradient descent works well, reaching desirable values for the objective function, as long as a diverse dataset of sufficient size is used. It also works regardless of the data used, as long as the objective function is differentiable w.r.t. the learnable parameters. This leads to a disadvantage; the gradient descent algorithms do not utilize any kind of information from the data, which could be used to find a better optima. To solve this issue, meta-learning was introduced.

Meta-learning in the SPD Manifold. In few words, meta-learning is considered to be an algorithm that learns how to learn [25]. In practice, it is a process, by means of an algorithm/framework, that adapts and learns to update the weights and biases of itself through a learning subsystem it contains. Intuitively, meta-learning is similar to the way humans adapt to new situations; they associate it to another similar task they know of, by modifying their learning approach based on the requirements at hand [25]. In the Euclidean space, Meta-learning has witness a number of implementations, using recurrent networks [3, 22, 51], reinforcement learning [26] or simple neural networks [6, 7]. They have proven to be invaluable tools when it comes to further enhance the ability of a system to perform a task, especially in cases where data are sparse for a given task, as they can leverage more information from them as stated earlier.

For the Riemannian SPD manifold, there were no formulations of a meta-learning framework until [17]. They proposed the matrix LSTM (mLSTM), a variant of the LSTM topology, that when given the gradient information $\nabla_{\mathbf{M}}$ of an SPD parameter $\mathbf{M}$, it can manipulate it without losing it's symmetric nature by utilizing the bilinear projection operation, $\mathrm{W}^{\mathrm{T}} \nabla_{\mathbf{M}} \mathrm{W}$. The mathematical formulation of the mLSTM is presented in Eq. (8) through Eq. (13):

$$I_{(t)} = \sigma\left(\mathrm{W}_{xi}^{\mathrm{T}} X_{(t)} \mathrm{W}_{xi} + \mathrm{W}_{hi}^{\mathrm{T}} H_{(t-1)} \mathrm{W}_{hi}\right) \quad (8)$$

$$F_{(t)} = \sigma\left(\mathrm{W}_{xf}^{\mathrm{T}} X_{(t)} \mathrm{W}_{xf} + \mathrm{W}_{hf}^{\mathrm{T}} H_{(t-1)} \mathrm{W}_{hf}\right) \quad (9)$$

$$O_{(t)} = \sigma\left(\mathrm{W}_{xo}^{\mathrm{T}} X_{(t)} \mathrm{W}_{xo} + \mathrm{W}_{ho}^{\mathrm{T}} H_{(t-1)} \mathrm{W}_{ho}\right) \quad (10)$$

$$\tilde{C}_{(t)} = \tanh\left(\mathrm{W}_{xc}^{\mathrm{T}} X_{(t)} \mathrm{W}_{xc} + \mathrm{W}_{hc}^{\mathrm{T}} H_{(t-1)} \mathrm{W}_{hc}\right) \quad (11)$$

$$C_{(t)} = F_{(t)} \odot C_{(t-1)} + I_{(t)} \odot \tilde{C}_{(t)} \quad (12)$$

$$H_{(t)} = O_{(t)} \odot \tanh\left(C_{(t)}\right) \quad (13)$$

where the subscript (t) denotes the time step, W_{ab} refers to the weight for parameter a at equation b, $\odot$ is the Hadamard product and $X_{(t)}$ refers to the symmetric data given to the network, which in the case of the meta-learning system is the gradient information, $\nabla_{\mathbf{M}}$, of an SPD parameter $\mathbf{M}$. The matrices $X, H,$ and C are of the same dimensionality d and symmetric, *i.e.*, $X, H, C \in Sym_d$, where Sym_d is the d-dimensional space of Symmetric matrices, and $W_{ab} \in \mathbb{R}^{d \times d}$. Thus, the first ever meta-learning subsystem has been formulated that operates with information from the SPD manifold, defined by the Eq. (14) through Eq. (21), in which $\mathbf{M}_{(t)}$ is the SPD parameter to be optimized. In detail, it utilizes two mLSTM networks, the first, mLSTM_s being responsible for the computation of the parameter update while the second, mLSTM_l computes the learning rate.

$$S_{l,(t-1)} = \left[H_{l,(t-1)}, C_{l,(t-1)}\right] \quad (14)$$

$$S_{s,(t-1)} = \left[H_{s,(t-1)}, C_{s,(t-1)}\right] \quad (15)$$

$$S_{(t-1)} = S_{l,(t-1)} \odot S_{s,(t-1)} \quad (16)$$

$$S_{l,(t)} = \text{mLSTM}_l\left(\nabla_{\mathbf{M}_{(t)}}, S_{(t-1)}\right) \quad (17)$$

$$S_{s,(t)} = \text{mLSTM}_s\left(\nabla_{\mathbf{M}_{(t)}}, S_{(t-1)}\right) \quad (18)$$

$$U_{(t)} = \pi_{\mathbf{M}_{(t)}}\left(\mathrm{W}_s^{\mathrm{T}}\left(H_{s,(t)} + \nabla_{\mathbf{M}_{(t)}}\right)\mathrm{W}_s\right) \quad (19)$$

$$\lambda_{(t)} = w_l^{\mathrm{T}} H_{l,(t)} w_l \quad (20)$$

$$\mathbf{M}_{(t+1)} = \mathrm{R}_{\mathbf{M}_{(t)}}\left(-\lambda_{(t)} U_{(t)}\right) \quad (21)$$

In order to optimize the set of the network parameters $\phi = \{\mathrm{W}_{xi}, \mathrm{W}_{hi}, \mathrm{W}_{xf}, \mathrm{W}_{hf}, \mathrm{W}_{xo}, \mathrm{W}_{ho}, \mathrm{W}_{xc}, \mathrm{W}_{hc}, \mathrm{w}_l, \mathrm{W}_s\}$, a two-part training algorithm is employed. It utilizes a set of m SPD parameters, $\mathbf{L}_{(t)} = \{\mathbf{L}_{(t),1}, \ldots, \mathbf{L}_{(t),m}\}$, where (t) denotes the parameter at time step t, that diminishes unwanted training oscillations and an experience pool, ψ, which stores previous knowledge and reused in every learning epoch. The first part of the training procedure, the observation stage, fills the experience pool with knowledge from a Riemannian Gradient Descent algorithm, along with the zero initialized network state $S_{(0)}$. In the learning stage, the network is used to optimize the set of SPD parameters for T steps, accumulating a global loss in each step. Once completed, the gradient of the global loss w.r.t to ϕ, $\nabla_\phi \mathcal{L}$ is computed and used to update the parameters of the network, using the ADAM algorithm. If τ optimization steps have been performed on any of the SPD parameters of $\mathbf{L}_{(t)}$, it and it's corresponding network state $S_{(t)}$, are reset to the identity matrix $\mathbf{I}_d$ and the zero filled matrix $\mathbf{0}_d$, where the subscript d denotes the rows and columns of the two aforementioned matrices. Lastly, the set $\mathbf{L}_{(t)}$ and the state $S_{(t)}$ are inserted into the experience pool, and the learning stage starts over.

5. Implementation

The proposed framework for WI-SV consists of two parts: the meta-learning network of paragraph 4 and a learnable covariance matrix $\mathbf{M}$ for the Mahalanobis distance in Eq. (22). We use the former to calculate the latter, by introducing a modification of the original training algorithm found in [17] in order to jointly train the optimizer and calculate $\mathbf{M}$. Algorithm 1 provides in detail the proposed system implementation: Specifically, we introduce an outer loop at the learning stage of the optimizer followed by another loop that calculates the covariance matrix. With this modification, we can invoke the learning stage multiple times, such that both the meta-learning network and the covariance parameter are improved in a similar rate. The hyperparameters chosen for the optimization network are also presented in a tabulated form in the supplementary. For the training process, we opt to keep an equal population of genuine-to-forgery and genuine-to-genuine samples so as to avoid any bias from the ω^+ and ω^- distributions. This was made by random sampling, while all the unpicked ones formed the validation set.

Loss metric. We utilize the contrastive loss of Eq. (23) as the objective function for filling the experience pool ψ and updating the covariance parameter $\mathbf{M}$. The objective function needs to be modified in a way that accommodates multiple calculations of it with multiple SPD parameters, which for our case, we can simply perform a mean value calculation of the loss function w.r.t the set of SPD parameters $\mathbf{L}_{(\mathbf{t})}$, as shown in Eq. (24).

$$d(x_{ij,kl}, \mathbf{M}_{(t)}) = \sqrt{x_{ij,kl}^{\mathrm{T}} \mathbf{M}_{(t)} x_{ij,kl}} \quad (22)$$

$$l(D, S, \mathbf{M}_{(t)}) = \frac{1}{|D|} \sum_{i,j,k,l \in D} (1 - \mathrm{y}_{ij,kl}) \max(0, \zeta_d - d)^2 + \frac{1}{|S|} \sum_{i,j,k,l \in S} \mathrm{y}_{ij,kl} \max(0, d - \zeta_s)^2 \quad (23)$$

$$\mathcal{L}(D, S, \mathbf{L}_{(\mathbf{t})}) = \frac{1}{m} \sum_{n=1}^{m} l(D, S, \mathbf{L}_{(t),n}) \quad (24)$$

where D, S are sets of samples from the ω^- and ω^+ distributions, $|D|, |S|$ are their cardinality, $x_{ij,kl}$ corresponds to the DP vector computed from the j-th and l-th samples of the i-th and k-th writer accordingly, $\mathrm{y}_{ij,kl}$ is the label of the corresponding DP vector which is 0 for dissimilar and 1 for similar pair of signatures, d in Eq. (23) refers to the Mahalanobis distance of Eq. (22), and $\mathbf{L}_{(t)}$ corresponds to the batched SPD parameters required for the meta-learning network's training process to reduced training oscillations.

6. Experiments & Results

Datasets & Setups. Experiments were conducted, with the use of two popular handwritten signature datasets of diverse origin. The first one is the CEDAR [24] dataset which consists of 55 writers with 24 genuine and 24 skilled forgery samples each. The second one is the Bengali part of the BHsig260 [39] (BHsig260-B) dataset, consisting of 100 writers with 24 genuine and 30 skilled forgery samples each. In all experiments, we learn the optimizer and the Mahalanobis distance with a development set (training & validation) followed by applying the testing set and measuring its efficiency. Both development as well testing sets are blind, *i.e.* they are disjoint. Having this in mind, we define two categories of dataset-oriented experiments. In the first, (*i.e.* blind-intra) a 5-by-2 fold cross validation (CV) is applied, meaning that on one fold, the development set is comprised by randomly selecting a 50% of the dataset writers, while the testing set is formed by the remaining 50%. For the same fold index, we next exchange the roles of development and testing set; and this is repeated 5 times. In the second, (*i.e.* blind-inter or cross-lingual), one dataset is considered as the development set while the other forms the testing set.

For both setups, we provide details on the formation of the development set, that is the S, D populations of Eq. (23)

Algorithm 1: Outline of the learning algorithm

Input: Rand. init. meta-learning optimizer parameters ϕ. Rand. init. Mahalanobis covariance parameter $\mathbf{M}_{(0)}$. SPD meta-learning network training batched SPD parameter $\mathbf{L}_{(0)} = \{\mathbf{I}_{d,1}, \ldots, \mathbf{I}_{d,m}\}$ and the corresponding mLSTM states $S_{(0)} = \{\mathbf{0}_{d,1}, \ldots, \mathbf{0}_{d,m}\}$. Experience pool $\psi = \varnothing$

Output: Mahalanobis covariance parameter $\mathbf{M}_{(t)}$

while $i \neq observation\_epochs$ **do**
  Compute l with $\mathbf{L}_{(t)}$ using Eq. (23) and $\nabla_{\mathbf{L}_{(t)}} l$.
  Compute $\mathbf{L}_{(t+1)}$ by Eq. (5).
  $\mathbf{L}_{(t)} \leftarrow \mathbf{L}_{(t+1)}$
  Insert $\{\mathbf{L}_{(t)}, S_{(0)}\}$ into ψ.
end
while $itr \neq learning\_iterations\ or\ convergence$ **do**
  while $oitr \neq optimizer\_learning\_epochs$ **do**
    Randomly sample $\{\mathbf{L}_{(t)}, S_{(t)}\}$ from ψ.
    while $step \neq T$ **do**
      Compute l with $\mathbf{L}_{(t)}$, $\nabla_{\mathbf{L}_{(t)}} l$ using Eq. (23).
      Compute $\mathbf{L}_{(t+1)}$ by Eq. (14) through Eq. (21).
    end
    Compute $\mathcal{L}$, $\nabla_{\phi}\mathcal{L}$ by Eq. (24).
    Update ϕ using the ADAM algorithm.
    if $t + T > \tau$ **then**
      $\mathbf{L}_{(t)} \leftarrow \{\mathbf{I}_{d,1}, \ldots, \mathbf{I}_{d,m}\}$
      $S_{(t)} =\leftarrow \{\mathbf{0}_{d,1}, \ldots, \mathbf{0}_{d,m}\}$
    end
    Insert $\{\mathbf{L}_{(t)}, S_{(t)}\}$ into ψ.
  end
  while $litr \neq learner\_epochs$ **do**
    Compute l with $\mathbf{M}_{(t)}$, $\nabla_{\mathbf{M}_{(t)}} l$ using Eq. (23).
    Compute $\mathbf{M}_{(t+1)}$ by Eq. (14) through Eq. (21).
  end
  Calculate validation metric $v_{(t)}$.
  if $v_{(t)} \approx v_{(t-1)}\ for\ converge\_epochs\ or\ v_{(t)} \geq 0.99$ **then**
    $convergence \leftarrow$ **True**.
  end
end
Return $\mathbf{M}_{(t)}$.

which correspond to the ω^{D+} and ω^{D-} classes. We designate with ω^{D+}_{WGG} the set of genuine-to-genuine signature pairs of any writer W that belongs to the development set. Thus, the ω^{D+} class is comprised by concatenating all the ω^{D+}_{WGG} subsets of all developments writers. Let us also define with ω^{D-}_{WGSF} and ω^{D-}_{WGRF} the genuine-to-skilled forgery set and the genuine-to-random forgery set as defined in the introduction. The ω^{D-} class of the development set is comprised by mixing these two subsets distributions in different ratios. Specifically, if $|D|$ is the cardinality of the ω^{D+} category, then we are fixing the cardinality of the ω^{D-} category $|S|$ equal to $|D|$ by randomly sampling an equal amount of genuine-to-forgery samples in order to not induce any bias.

In this work, the ω^{D-} category is formed under three setups: In the first setup, named as (0%RF), the ω^{D-} is comprised only by ω^{D-}_{WGSF} samples, in the (50%RF) the ω^{D-} is comprised by a equal percentage of ω^{D-}_{WGSF} and ω^{D-}_{WGRF} and finally in the (100%RF) the ω^{D-} is comprised only by ω^{D-}_{WGRF} samples. For the blind-intra case, we partition the development set of $\{\omega^{D+}, \omega^{D-}\}$ to the training and validation sets $\{\omega^{T+}, \omega^{T-}\}$ and $\{\omega^{V+}, \omega^{V-}\}$. Their cardinality values, $|T|$ and $|V|$ were set to: $|T| = 70\%|S|$ and $|V| = 30\%|S|$. For the cross-lingual case, the cardinality values, $|T|$ and $|V|$ were set to: $|T| = 90\%|S|$ and $|V| = 10\%|S|$.

The development set is employed in order to learn the covariance matrix $\mathbf{M}$ of Eq. (22). For each writer of the testing set, in both intra-or-inter-blind setups, we randomly select 10 genuine samples as references. The rest of the genuine along with the skilled forgeries are then used as questioned samples which form pairs with the ten references. For each questioned sample, we compute its corresponding distances and then, we select the minimum distance in order to assign it a score. These scores, conditioned as genuine or forgery are then used to calculate the writer's Equal Error Rate (EER%) through its Receiver Operating Characteristic (ROC) curve, by finding the point in the curve where the False Positive Rate and the False Negative Rate are equal. We repeat this procedure 10 times and compute a mean, user level EER%. Both for the blind-intra and cross-lingual cases, we report the error rates by utilizing the three different setups, mentioned to construct the negative distribution of samples. In Table 2 we present the results, where the row corresponds to the dataset and DP used for training, while the column to the Random Forgery % (RF%) and dataset used for testing. Cross-lingual testing is the result of the latter training process, while intra-dataset EER% results are calculated from the 5-by-2 CV procedure.

Inspecting Table 2, we notice that the DP^{NOL} representation of BHsig260-B seems to have the best performance, in both intra- and inter-blind-setups, when using 50% RF for the ω^{-} population, while for CEDAR, the best intra dataset result is seen at the 50% RF mix as well, but the best cross-lingual is seen in 0% RF. Following a similar pattern, the best performance on both testing categories of BHsig260-B DP^{OL} is again at the 50% mix, while for CEDAR's overlapping DP vector is seen at the 0% RF. Al-

RF%	0%		50%		100%	
Training models	BHsig260-B	CEDAR	BHsig260-B	CEDAR	BHsig260-B	CEDAR
BHsig260-B: DP^{NOL}	0.14	0.09	0.11	0.09	0.11	0.13
BHsig260-B: DP^{OL}	0.08	0.09	0.07	**0.02**	0.07	0.05
CEDAR: DP^{NOL}	0.08	0.12	0.11	0.08	0.13	0.09
CEDAR: DP^{OL}	**0.03**	0.05	0.06	0.06	0.08	0.08

Table 2. Results on blind-intra- and cross-lingual-dataset testing EER (%) under the three different RF(%) setups.

1st Author #Ref	Method / Features	# Refs.	CEDAR	BHsig260-B
Liu [28]	DCSM / MSDN	10	1.75	-
Hamadene [19]	FDMT / CT	5	2.10	-
Parcham [40]	CBCapsNet / CBCapsNet	N/A	0	5.70
Hanif [21]	Mahalanobis Distance / HOG + LBP + VLAD	N/A	0	9.62
Chattopadhyay [12]	ResNet-18 / SURDS	N/A	-	12.6
Lin [27]	Euclidean Distance / 2C2L	N/A	0	11.92
Lu [31]	ADF / Spatial Transformer	All	0	3.96
Natarajan [37]	Euclidean Distance / SigNet	N/A	0	4.38
Viana [49]	Euclidean Distance / MHSRL	10	4.91	-
Wan [50]	Euclidean Distance /SigCNN	1	0	-
Ren [42]	Euclidean Distance / SigGCN	5 / 20	0	4.00
Huang [23]	Euclidean Distance / MGRNet	All	3.51	6.12
Ours, intra 100 RF%	Mahalanobis Distance / DP^{NOL}	10	0.09	0.11
Ours, inter 100 RF%	Mahalanobis Distance / DP^{NOL}	10	0.13	0.13
Ours, intra 100 RF%	Mahalanobis Distance / DP^{OL}	10	0.08	0.07
Ours, inter 100 RF%	Mahalanobis Distance / DP^{OL}	10	0.05	0.08

Table 3. Summary of State-of-the-Art results for metric learning SV systems with ours on the reported EER (%) metric.

though the eleven dimensional overlapping representation performs better, the non-overlapping one performs remarkably well for a three element representation. Even so, DP^{OL} contains more information which is the most probable reason as to why it performs better. In all cases, the error rates are quite low, presenting minor differences with each other. This could be the result of the random initialization of the parameters, as some may have led to better optimas, but weight initialization analysis and optimal parameterization of the distribution used for the random initialization is out of the scope of this paper.

Table 3 delivers the results of the 100% RF tests as a part to a comparative summary of the WI-SV methods with metric learning approaches found in the literature. We choose to present the 100% RF population as we anticipate that when an SV system is used outside of research environment, it would be hard to collect skilled forgery samples. Motivated by the low verification error results, we feel that our proposed method can be considered, at least, to be a performant approach to the challenging task of WI-SV. We run our experiments on a NVIDIA-DGX equipped with four V100 GPU's along with an RTX3060 Ti GPU.

7. Conclusion

In this work, we addressed writer independent signature verification in the space of Symmetric Positive Definite matrices under a metric meta-learning concept. First, pairs of signatures were mapped to a vector with elements SPD distances, termed as the Distance Pyramid. These vectors are used as the development data for the meta-learning signature verifier. The proposed system learns a covariance parameter $\mathbf{M}$, of a Mahalanobis distance, and distinguishes similar from dissimilar pairs, where the former must be close to each other while the latter must be far away. The experiments were performed with two datasets of diverse origin, in a blind intra and blind inter manner, with the results indicating that our method is performant and efficient, with a small number of parameters. Future work entails more datasets and perhaps a meta-learning framework for optimal parameterization and many other DP structures in order to create more descriptive representations that advance signature verifiers.

References

[1] P. A. Absil, Robert Mahoney, and Rodolphe Sepulchre. Optimization Algorithms on Matrix Manifolds, 1999. 4, 5

[2] Linda C. Alewijnse. Analysis of Signature Complexity, Master Thesis. Master's thesis, University of Amsterdam, 2008. Volume: Master Thesis. 1

[3] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent, Nov. 2016. arXiv:1606.04474 [cs]. 5

[4] Colin Atkinson and Ann F. S. Mitchell. Rao's Distance Measure. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 43(3):345–365, 1981. 3, 4

[5] Seth D. Axen, Mateusz Baran, Ronny Bergmann, and Krzysztof Rzecki. Manifolds.jl: An Extensible Julia Framework for Data Analysis on Manifolds, June 2023. arXiv:2106.08777 [cs]. 5

[6] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the Optimization of a Synaptic Learning Rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, 2007. 5

[7] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. Learning a synaptic learning rule. In *IJCNN-91-Seattle International Joint Conference on Neural Networks*, volume ii, page 969, Seattle, WA, USA, 1991. IEEE. 5

[8] Michael. Blumenstein, Miguel A. Ferrer, and J. Francisco Vargas. The 4NSigComp2010 Off-line Signature Verification Competition: Scenario 2. In *2010 12th International Conference on Frontiers in Handwriting Recognition*, pages 721–726, Nov. 2010. 3

[9] Walid Bouamra, Chawki Djeddi, Brahim Nini, Moises Diaz, and Imran Siddiqi. Towards the design of an offline signature verifier based on a small number of genuine samples for training. *Expert Systems with Applications*, 107:182–195, Jan. 10. 1

[10] Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, Cambridge ; New York, NY, 2023. 4, 5

[11] Nicolas Boumal, Bamdev Mishra, P. A. Absil, and Rodolphe Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 2013. Publisher: arXiv Version Number: 1. 5

[12] Soumitri Chattopadhyay, Siladittya Manna, Saumik Bhattacharya, and Umapada Pal. SURDS: Self-Supervised Attention-guided Reconstruction and Dual Triplet Loss for Writer Independent Offline Signature Verification. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1600–1606, Aug. 2022. 3, 8

[13] Marcos Faundez-Zanuy, Julian Fierrez, Miguel A. Ferrer, Moises Diaz, Ruben Tolosana, and Réjean Plamondon. Handwriting Biometrics: Applications and Future Trends in e-Security and e-Health. *Cognitive Computation*, 12(5):940–953, Sept. 2020. 1

[14] Miguel Angel Ferrer, Moises Diaz-Cabrera, and Aythami Morales. Static Signature Synthesis: A Neuromotor Inspired Approach for Biometrics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):667–680, 2015. 3

[15] Miguel Angel Ferrer, Jesús Francisco Vargas, Aythami Morales, and Aarón Ordonez. Robustness of Offline Signature Verification Based on Gray Level Features. *IEEE Trans. Forens. and Secur.*, 7(3):966–977, 2012. 3

[16] Javier Galbally, Moises Diaz-Cabrera, Miguel A. Ferrer, Marta Gomez-Barrero, Aythami Morales, and Julian Fierrez. On-line signature recognition through the combination of real dynamic data and synthetically generated static data. *Pattern Recognition*, 48(9):2921–2934, 9. 3

[17] Zhi Gao, Yuwei Wu, Yunde Jia, and Mehrtash Harandi. Learning to Optimize on SPD Manifolds. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7697–7706, June 2020. 2, 5, 6

[18] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *IEEE Conf. Comput. Vis. Pattern Recog.*, volume 2, pages 1735–1742, 2006. 2

[19] Assia Hamadene and Youcef Chibani. One-Class Writer-Independent Offline Signature Verification Using Feature Dissimilarity Thresholding. *IEEE Trans. Forens. and Secur.*, 11(6):1226–1238, 2016. 3, 8

[20] Muzaffar M. Hameed, Rodina Ahmad, Miss Laiha Mat Kiah, and Ghulam Murtaza. Machine learning-based offline signature verification systems: A systematic review. *Signal Processing: Image Communication*, 93:116139, Apr. 2021. 1, 2

[21] Muhammad Shehzad Hanif and Muhammad Bilal. A Metric Learning Approach for Offline Writer Independent Signature Verification. *Pattern Recognition and Image Analysis*, 30(4):795–804, Oct. 2020. 3, 8

[22] Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. Learning to Learn Using Gradient Descent. In Georg Dorffner, Horst Bischof, and Kurt Hornik, editors, *Artificial Neural Networks — ICANN 2001*, Lecture Notes in Computer Science, pages 87–94, Berlin, Heidelberg, 2001. Springer. 5

[23] Fu-Hsien Huang and Hsin-Min Lu. Multiscale global and regional feature learning using co-tuplet loss for offline handwritten signature verification, 2023. 3, 8

[24] Meenakshi K. Kalera, Sargur Srihari, and Aihua Xu. Offline signature verification and identification using distance statistics. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(07):1339–1360, Nov. 2004. 3, 6

[25] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review*, 44(1):117–130, June 2015. 5

[26] Ke Li and Jitendra Malik. Learning to Optimize, June 2016. arXiv:1606.01885 [cs, math, stat]. 5

[27] Feng Lin, Chuang Li, Zhiyong Wang, Gang Yu, Liou Yuan, and Haiqiang Wang. DeepHSV: User-Independent Offline Signature Verification Using Two-Channel CNN. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 166–171, Sept. 2019. ISSN: 2379-2140. 3, 8

[28] Li Liu, Linlin Huang, Fei Yin, and Youbin Chen. Offline signature verification using a region based deep metric learning network. *Pattern Recognition*, 118:108009, Oct. 2021. 3, 8

[29] Marcus Liwicki, Muhammad Imran Malik, C. Elisa Van Den Heuvel, Xiaohong Chen, Charles Berger, Reinoud Stoel, Michael Blumenstein, and Bryan Found. Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011). In *2011 International conference on document analysis and recognition*, pages 1480–1484, Sept. 2011. 3

[30] Teressa Longjam, Dakshina Ranjan Kisku, and Phalguni Gupta. Writer independent handwritten signature verification on multi-scripted signatures using hybrid CNN-BiLSTM: A novel approach. *Expert Systems with Applications*, 214:119111, Mar. 2023. 1

[31] Xi Lu, Linlin Huang, and Fei Yin. Cut and Compare: End-to-end Offline Signature Verification Network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3589–3596, Jan. 2021. 3, 8

[32] Paul Maergner, Nicholas Howe, Kaspar Riesen, Rolf Ingold, and Andreas Fischer. Offline Signature Verification Via Structural Methods: Graph Edit Distance and Inkball Models. In *16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 163–168, Aug. 2018. 3

[33] Paul Maergner, Vinaychandran Pondenkandath, Michele Alberti, Marcus Liwicki, Kaspar Riesen, Rolf Ingold, and Andreas Fischer. Combining graph edit distance and triplet networks for offline signature verification. *Pattern Recognition Letters*, 125:527–533, July 2019. 3

[34] Muhammad Imran Malik and Marcus Liwicki. From Terminology to Evaluation: Performance Assessment of Automatic Signature Verification Systems. In *2012 International Conference on Frontiers in Handwriting Recognition*, pages 613–618, Sept. 2012. 1

[35] Saeed Masoudnia, Omid Mersa, Babak Nadjar Araabi, Abdol-Hossein Vahabie, Mohammad Amin Sadeghi, and Majid Nili Ahmadabadi. Multi-Representational Learning for Offline Signature Verification using Multi-Loss Snapshot Ensemble of CNNs. *Expert Systems with Applications*, Mar. 2019. 1

[36] Moises Diaz, Miguel A. Ferrer, Donato Impedovo, Muhammad Imran Malik, Giuseppe Pirlo, and Réjean Plamondon. A Perspective Analysis of Handwritten Signature Technology. *ACM Comput. Surv. 1, 1, Article 1 (January 2018)*, 2018. 1

[37] Abhiram Natarajan, B. Sathish Babu, and Xiao-Zhi Gao. Signature warping and greedy approach based offline signature verification. *International Journal of Information Technology*, May 2021. 3, 8

[38] J. Ortega-Garcia, J. Fierrez-Aguilar, D. Simon, J. Gonzalez, M. Faundez-Zanuy, V. Espinosa, A. Satue, I. Hernaez, J.-J. Igarza, C. Vivaracho, D. Escudero, and Q.-I. Moro. MCYT baseline corpus: a bimodal biometric database. *IEE Proceedings - Vision, Image and Signal Processing*, 150(6):395–401, Dec. 2003. Publisher: IET Digital Library. 3

[39] Srikanta Pal, Alireza Alaei, Umapada Pal, and Michael Blumenstein. Performance of an Off-Line Signature Verification Method Based on Texture Features on a Large Indic-Script Signature Dataset. In *12th IAPR workshop on document analysis systems (DAS)*, pages 72–77, Apr. 2016. 3, 6

[40] Ebrahim Parcham, Mahdi Ilbeygi, and Mohammad Amini. CBCapsNet: A novel writer-independent offline signature verification model using a CNN-based architecture and capsule neural networks. *Expert Systems with Applications*, 185:115649, Dec. 2021. 3, 8

[41] Elżbieta Pękalska and Robert P. W. Duin. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23(8):943–956, 2002. 1

[42] Chengkai Ren, Jian Zhang, Hongwei Wang, and Shuguang Shen. Vision Graph Convolutional Network for Writer-Independent Offline Signature Verification. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, June 2023. ISSN: 2161-4407. 3, 8

[43] Amir Soleimani, Babak N. Araabi, and Kazim Fouladi. Deep Multitask Metric Learning for Offline Signature Verification. *Pattern Recognition Letters*, 80:84–90, 2016. 3

[44] Amir Soleimani, Kazim Fouladi, and Babak N. Araabi. UTSig: A Persian offline signature dataset. *IET Biometrics*, 6(1):1–8, 2016. 1, 3

[45] Victor L. F. Souza, Adriano L. I. Oliveira, Rafael M. O. Cruz, and Robert Sabourin. A white-box analysis on the writer-independent dichotomy transformation applied to offline handwritten signature verification. *Expert Systems with Applications*, 154:113397, 2020. 1

[46] Sargur N. Srihari, Sung-Hyuk Cha, Hina Arora, and Sangjik Lee. Individuality of handwriting. *Journal of forensic science*, 47(4):1–17, 2002. 1

[47] Sargur N. Srihari, Aihua Xu, and Meenakshi K. Kalera. Learning strategies and classification methods for off-line signature verification. In *Ninth International Workshop on Frontiers in Handwriting Recognition*, pages 161–166, 2004. 1

[48] Michael Stauffer, Paul Maergner, Andreas Fischer, and Kaspar Riesen. A Survey of State of the Art Methods Employed in the Offline Signature Verification Process. In Rolf Dornberger, editor, *New Trends in Business Information Systems and Technology: Digital Innovation and Digital Business Transformation*, pages 17–30. Springer International Publishing, Cham, 2021. 1

[49] Talles B. Viana, Victor L. F. Souza, Adriano L. I. Oliveira, Rafael M. O. Cruz, and Robert Sabourin. A multi-task approach for contrastive learning of handwritten signature feature representations. *Expert Systems with Applications*, 217:119589, May 2023. 3, 8

[50] Qian Wan and Qin Zou. Learning Metric Features for Writer-Independent Signature Verification using Dual Triplet Loss. In *2020 25th international conference on pattern recognition (ICPR)*, pages 3853–3859, Jan. 2021. 3, 8

[51] Olga Wichrowska, Niru Maheswaranathan, Matthew W. Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando de Freitas, and Jascha Sohl-Dickstein. Learned Optimizers that Scale and Generalize, Sept. 2017. arXiv:1703.04813 [cs, stat]. 5

[52] Yecheng Zhu, Songxuan Lai, Zhe Li, and Lianwen Jin. Point-to-Set Similarity Based Deep Metric Learning for Offline Signature Verification. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 282–287, Sept. 2020. 3

[53] Elias N. Zois, Alex Alexandridis, and George Economou. Writer independent offline signature verification based on asymmetric pixel relations and unrelated training-testing datasets. *Expert Systems with Applications*, 125:14–32, July 2019. 1

[54] Elias N. Zois, Salem Said, Dimitrios Tsourounis, and Alex Alexandridis. Subscripto multiplex: A Riemannian symmetric positive definite strategy for offline signature verification. *Pattern Recognition Letters*, 167:67–74, Mar. 2023. 2

[55] Elias N. Zois, Dimitrios. Tsourounis, Ilias Theodorakopoulos, Aanastasios L. Kesidis, and George Economou. A Comprehensive Study of Sparse Representation Techniques for Offline Signature Verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 1(1):68–81, 2019. 2