DreamNet: A Deep Riemannian Manifold Network for SPD Matrix Learning

Rui Wang[1,2][0000−0002−9984−1752], Xiao-Jun Wu[1,2⋆][0000−0002−0310−5778], Ziheng Chen[1,2][0000−0002−5366−7293], Tianyang Xu[1,2][0000−0002−9015−3128], and Josef Kittler[1,2,3][0000−0002−8110−9205]

[1] School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China
[2] Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China
[3] Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, Guildford GU2 7XH, U.K.
cs_wr@jiangnan.edu.cn, zh_chen@stu.jiangnan.edu.cn, j.kittler@surrey.ac.uk
{xiaojun_wu_jnu,tianyang_xu}@163.com

Abstract. The methods of symmetric positive definite (SPD) matrix learning have attracted considerable attention in many pattern recognition tasks, as they are eligible to capture and learn appropriate statistical features while respecting the Riemannian geometry of SPD manifold where the data reside on. Accompanied with the advanced deep learning techniques, several Riemannian networks (RiemNets) for SPD matrix nonlinear processing have recently been studied. However, it is pertinent to ask, whether greater accuracy gains can be realized by simply increasing the depth of RiemNets. The answer appears to be negative, as deeper RiemNets may be difficult to train. To explore a possible solution to this issue, we propose a new architecture for SPD matrix learning. Specifically, to enrich the deep representations, we build a stacked Riemannian autoencoder (SRAE) on the tail of the backbone network, *i.e.*, SPDNet [23]. With this design, the associated reconstruction error term can prompt the embedding functions of both SRAE and of each RAE to approach an identity mapping, which helps to prevent the degradation of statistical information. Then, we implant several residual-like blocks using shortcut connections to augment the representational capacity of SRAE, and to simplify the training of a deeper network. The experimental evidence demonstrates that our DreamNet can achieve improved accuracy with increased depth.

Keywords: SPD Matrix Learning · Riemannian Neural Network · Information Degradation · Stacked Riemannian Autoencoder (SRAE)

1 Introduction

Covariance matrices are well-known in any statistical-related field, but their direct usage as data descriptors in the community of computer vision and pattern

⋆ Corresponding author

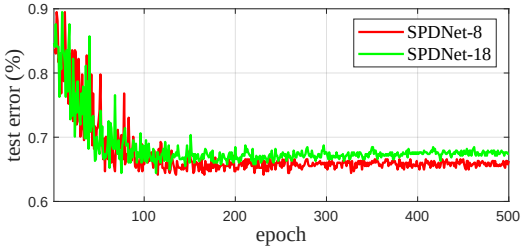


Fig. 1. Test error of SPDNets versus the number of epochs on the AFEW dataset

recognition (CV&PR) is less common. Even so, their effectiveness has been verified in a variety of applications. In medical imaging, covariance matrices are taken to classify time-series for Brain-Computer Interfaces (BCI) [3] and analyze magnetic resonance imaging (MRI) [8, 5]. In visual classification, since a global covariance matrix has the capacity to characterize the spatiotemporal fluctuations of data points of different lengths, covariance features have gained remarkable progress in many practical scenarios, such as dynamic scene classification [38, 46, 47], facial emotional recognition [23, 4, 44], face recognition [26, 24, 19], and action recognition [18, 33, 51], *etc.*

However, the main difficulty of processing and classifying these matrices, which are actually SPD, is that they cannot be regarded as the Euclidean elements, as their underlying space is a curved Riemannian manifold, *i.e.*, an SPD manifold [2]. Consequently, the tools from Euclidean geometry cannot directly be applied for computation. Thanks to the well-studied Riemannian metrics, including Log-Euclidean Metric (LEM) [2] and Affine-Invariant Riemannian Metric (AIRM) [34], the Euclidean methods can be generalized to the SPD manifolds by either mapping it into an associated flat space via tangent approximation [41, 40, 36] or utilizing the Riemannian kernel functions to embed it into a Reproducing Kernel Hilbert Space (RKHS) [48, 43, 20, 17]. However, these two types of approaches may lead to undesirable solutions as they distort the geometrical structure of the input data manifold by the data transformation process. To respect the original Riemannian geometry more faithfully, several geometry-aware discriminant analysis algorithms [26, 54, 19, 13] have been developed for learning an efficient, manifold-to-manifold projection mapping. Regrettably, despite their notable success, the intrinsic shallow linear SPD matrix learning scheme, implemented on nonlinear manifolds, impede these methods from mining fine-grained geometric representations.

Motivated by the philosophy of convolutional neural networks (ConvNets) [21, 37], an end-to-end Riemannian architecture for SPD matrix nonlinear learning has been proposed (SPDNet [23]). The structure of SPDNet is analogous to a classical ConvNet (*e.g.*, with transformation and activation layers), but each layer processes the SPD manifold-valued data points. The final layer of SPDNet maps the learned feature manifold into a flat space for classification. More architectures have followed thereafter [4, 46, 33, 51], modifying the elementary building blocks for different application scenarios. As recent evidence [37, 21] reveals, the network depth is of vital importance for promoting good perfor-

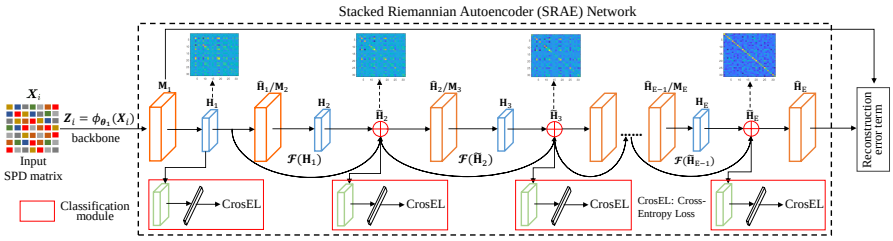


Fig. 2. Schematic diagram of the proposed Riemannian network.

mance. A question therefore arises: *can the classification accuracy be improved by simply stacking more layers on top of each other in the SPD neural networks?* The following three factors make it impossible to provide ready answers: 1) existing RiemNets have a small number of layers and there is no prior experience in building very deep RiemNets; 2) there is limited research on this topic; 3) deeper SPD network may be difficult to train. A typical example is illustrated in Fig. 1. It should be noted that the classification error of SPDNet-18 is higher than that of SPDNet-8.

The above observation suggests that simply stacking more layers on top of each other does not mean that a better RiemNet can be learnt. This article proposes a new architecture for SPD matrix processing and classification that avoids the pitfalls of layer stacking in RiemNet. The overall framework of our approach is shown in Fig. 2. As a greater depth of representation is essential for many classification tasks[49, 55, 21, 53], the purpose of the proposed network is to pursue a deeper manifold-to-manifold embedding mapping that would transform the input SPD matrices into more informative ones of lower dimensionality and the same topology. To meet this requirement, we select the original architecture proposed in [23] as the backbone of our model, in view of its demonstrable strength in SPD matrix nonlinear learning. Then, a stacked Riemannian autoencoder network (SRAE) is established at the end of the backbone to increase the depth of the structured representations. Under the supervision of a reconstruction error term associated with the input-output SPD matrices of SRAE, the embedding mechanisms of both SRAE and each RAE will asymptotically approach an identity mapping, thus being capable of preventing a degradation of statistical information during multi-stage data compressed sensing. The proposed solution ensures that the classification error produced by our deeper model would not be higher than that of the shallower backbone. To enhance the representational capacity of SRAE, we build multiple residual-like blocks within it, implemented by the shortcut connections [21] between the hidden layers of any two adjecent RAEs. This design makes the current RAE learning stage access the informative features of the previous stages easily, facilitating the reconstruction of the remaining structural details. Since the above design ensures that the SRAE network remains sensitive to the data variations in the new feature manifolds, we also append a classification module, composed of the LogEig layer (will be introduced later), FC layer, and cross-entropy loss, to each RAE to facilitate the training of a discriminative manifold-to-manifold deep transformation map-

ping. In this manner, a series of effective classifiers can be obtained. Finally, a simple maximum voting strategy is applied for decision making.

We demonstrate the benefits of the proposed approach on the tasks of facial emotion recognition, skeleton-based hand action recognition, and skeleton-based action recognition with UAVs, respectively. The experimental results achieved on three benchmarking datasets show that our DreamNet achieves accuracy gains from an increasing network depth, producing better results than the previous methods.

2 Related Works

To endow SPD matrix representation learning with deep and nonlinear function, Ionescu et al. [28] integrate global SPD computation layers with the proposed matrix backpropagation methodology into deep networks to capture structured features for visual scene understanding. Inspired by the paradigm of ConvNets, Huang et al. [23] design a novel Riemannian neural network for SPD matrix nonlinear learning, comprising of a stack of SPD matrix transformation and activation layers, referred to as SPDNet. To provide a better guidance for the network training, Brooks et al. [4] design a Riemannian batch normalization module for SPDNet. Considering the potential importance of the local structural information contained in the SPD matrix, Zhang et al. [51] propose an SPD matrix 2D convolutional layer for data transformation, requiring each convolutional kernel also to be SPD. Different from [51], Chakraborty et al. [5] use the weighted Fréchet Mean (wFM) operation to simulate convolution on the manifolds, considering the intrinsic Riemannian geometry of the data points like diffusion tensors. More recently, Wang et al. [46] design a lightweight cascaded neural network for SPD matrix learning and classification, which shows higher computational efficiency and competitive classification performance, especially with limited training data.

3 Proposed Method

Although the Riemannian neural network approaches for SPD matrix processing can alleviate the negative impact of data variations on the classification performance, achieving accuracy gains is not simply a matter of increasing the network depth. The main obstacle to this simplistic solution is the degradation of statistical information (degradation problem), which makes the learned deep representations unable to effectively characterize the structural information of the original imaged scene, thus resulting in lower accuracy. In this paper, we design a novel Riemannian architecture named DreamNet to solve this issue. Fig. 2 provides an overview of our approach.

3.1 Preliminaries

SPD Manifold: A real-valued symmetric matrix $\boldsymbol{X}$ is called SPD if and only if $\boldsymbol{v}^T\boldsymbol{X}\boldsymbol{v} > 0$ for all non-zero vector $\boldsymbol{v} \in \mathbb{R}^d$. As studied in [2, 34], when endowed

with manifold structures, the set of d-by-d SPD matrices, denoted as $\boldsymbol{\mathcal{S}}_{++}^{d}$:

$$\boldsymbol{\mathcal{S}}_{++}^{d} := \{\boldsymbol{X} \in \mathbb{R}^{d\times d} : \boldsymbol{X} = \boldsymbol{X}^T, \boldsymbol{v}^T\boldsymbol{X}\boldsymbol{v} > 0, \forall \boldsymbol{v} \in \mathbb{R}^d \backslash \{0_d\}\}. \tag{1}$$

forms a specific Riemannian manifold, *i.e.*, SPD manifold. This enables the use of concepts related to differential geometry to address $\boldsymbol{\mathcal{S}}_{++}^{d}$, such as geodesic.

Data Modeling with Second-Order Statistics: Let $\boldsymbol{S}_i = [\boldsymbol{s}_1, \boldsymbol{s}_2, ..., \boldsymbol{s}_{n_i}]$ be the i^{th} given data sequence with n_i entries, where $\boldsymbol{s}_t \in \mathbb{R}^{d\times 1}$ denotes the t^{th} vectorized instance. For $\boldsymbol{S}_i$, its second-order representation is computed by: $\boldsymbol{X}_i = \frac{1}{n_i-1}\sum_{t=1}^{n_i}(\boldsymbol{s}_t - \boldsymbol{u}_i)(\boldsymbol{s}_t - \boldsymbol{u}_i)^T$, where $\boldsymbol{u}_i = \frac{1}{n_i}\sum_{t=1}^{n_i}\boldsymbol{s}_t$ signifies the mean of $\boldsymbol{S}_i$. Considering that $\boldsymbol{X}_i$ does not necessarily satisfy the condition of positive definiteness, it is regularised, *i.e.*, $\boldsymbol{X}_i \leftarrow \boldsymbol{X}_i + \lambda\boldsymbol{I}_d$, where $\boldsymbol{I}_d$ is an identity matrix of size $d \times d$, and λ is set to $trace(\boldsymbol{X}_i) \times 10^{-3}$ in all the experiments. In this way, $\boldsymbol{X}_i$ is a true SPD manifold-valued element [34].

Basic Layers of SPDNet: Let $\boldsymbol{X}_{k-1} \in \boldsymbol{\mathcal{S}}_{++}^{d_{k-1}}$ be the input SPD matrix of the k^{th} layer. The Riemannian operation layers defined in [23] are as follows:

BiMap Layer: This layer is analogous to the usual dense layer, used to transform the input SPD data points into a lower dimensional space by a bilinear mapping f_b, expressed as $\boldsymbol{X}_k = f_b^{(k)}(\boldsymbol{W}_k, \boldsymbol{X}_{k-1}) = \boldsymbol{W}_k^T\boldsymbol{X}_{k-1}\boldsymbol{W}_k$, where $\boldsymbol{W}_k$ is the column full-rank transformation matrix with semi-orthogonality.

ReEig Layer: This layer is similar to the classical ReLU layers, designed to inject nonlinearity into SPDNet by modifying the small positive eigenvalues of each input SPD matrix with a nonlinear rectification function f_r, formulated as $\boldsymbol{X}_k = f_r^{(k)}(\boldsymbol{X}_{k-1}) = \boldsymbol{U}_{k-1}\max(\epsilon\boldsymbol{I}, \boldsymbol{\Sigma}_{k-1})\boldsymbol{U}_{k-1}^T$. Here, $\boldsymbol{X}_{k-1} = \boldsymbol{U}_{k-1}\boldsymbol{\Sigma}_{k-1}\boldsymbol{U}_{k-1}^T$ represents the eigenvalue decomposition, and ϵ is a small activation threshold.

LogEig Layer: This layer is designed to perform the following logarithmic mapping: $\boldsymbol{X}_k = f_l^{(k)}(\boldsymbol{X}_{k-1}) = \boldsymbol{U}_{k-1}\log(\boldsymbol{\Sigma}_{k-1})\boldsymbol{U}_{k-1}^T$, where $\log(\boldsymbol{\Sigma})$ represents the logarithm operation on each diagonal element of $\boldsymbol{\Sigma}$, and $\boldsymbol{X}_{k-1} = \boldsymbol{U}_{k-1}\boldsymbol{\Sigma}_{k-1}\boldsymbol{U}_{k-1}^T$ denotes the eigenvalue decomposition. In the resulting flat space, the classification tasks can be realized with the conventional dense layers.

3.2 Deep Riemannian Network

As shown in Fig. 2, the designed SRAE module contains a cascade of Riemannian autoencoders (RAEs) to achieve continuous incremental reconstruction learning, in which the output feature maps of each RAE are used as the input data points of the adjacent one. To enrich the information flow in the SRAE network, we augment the sequential connections between adjacent RAEs using the shortcut connections, so that the current RAE module can effectively mine the revelant structural information with the aid of the former prediction for a better reconstruction. The network structure of each RAE is composed of three components. The first part is an encoder module, made up of the input (BiMap), nonlinear activation (ReEig), and hidden (BiMap) layers for geometry-aware dimensionality reduction of SPD matrices. The second part is the decoder module, mainly used for data reconstruction. Since it has a symmetric structure with the encoder, the

RAE is defined strictly in the context of Riemannian manifolds, and so is SRAE and the whole network. Moreover, each RAE also connects to a classification network with the layers of LogEig and FC, guided by the cross-entropy loss.

Let $\mathfrak{S} = [\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_N]$ and $\boldsymbol{L} = [l_1, l_2, ..., l_N] \in \mathbb{R}^{1\times N}$ be the original training set and its corresponding label vector, respectively. In this article, we denote the SPD manifold-valued training set as: $\boldsymbol{\mathcal{X}} = [\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_N]$. For the i^{th} input SPD matrix $\boldsymbol{X}_i$ of our DreamNet, the low-dimensional and compact feature matrix output by the backbone can be expressed as: $\boldsymbol{\mathcal{Z}}_i = \phi_{\boldsymbol{\theta}_1}(\boldsymbol{X}_i)$. Here, $\phi_{\boldsymbol{\theta}_1}$ represents the Riemannian network embedding from the input data manifold to the target one, realized by a stack of BiMap and ReEig layers. Besides, $\boldsymbol{\theta}_1$ indicates the to-be-learnt parameters of this backbone network. As the SRAE module consists of E RAEs, we use $\mathbf{M}_e$ ($\mathbf{M}_e = \boldsymbol{\mathcal{Z}}_i$ when $e = 1$), $\mathbf{H}_e$, and $\hat{\mathbf{H}}_e$ to denote the input, output of the hidden layer, and reconstruction of the input of the e^{th} ($e = 1 \rightarrow$ E) RAE, respectively. Thus, $\mathbf{H}_e$ and $\hat{\mathbf{H}}_e$ can be computed by:

$$\mathbf{H_e} = f_{b_e}(\boldsymbol{W}_{e_1}, \mathbf{M}_e) = \boldsymbol{W}_{e_1}^T \mathbf{M}_e \boldsymbol{W}_{e_1}, \tag{2}$$

$$\hat{\mathbf{H}}_e = f_{b_e}(\boldsymbol{W}_{e_2}, \mathbf{H}_e) = \boldsymbol{W}_{e_2} \mathbf{H}_e \boldsymbol{W}_{e_2}^T, \tag{3}$$

where f_{b_e} and $\boldsymbol{W}_{e_1} \in \mathbb{R}^{d_{e-1}\times d_e}$ ($d_e \leq d_{e-1}$), $\boldsymbol{W}_{e_2} \in \mathbb{R}^{d_{e-1}\times d_e}$ represent the bilinear mapping function and the transformation matrices of the e^{th} RAE, respectively. Since $\mathbf{M}_e$ is actually equivalent to $\hat{\mathbf{H}}_{e-1}$, we replace $\mathbf{M}_e$ with $\hat{\mathbf{H}}_{e-1}$ in the following for clarity.

Based on the constructed SRAE architecture, the shortcut connections (SCs) and element-wise addition (EWA) enable the Riemannian residual learning to be adopted for every set of a few stacked layers. In this article, we define the building block shown in Fig. 2 as:

$$\tilde{\mathbf{H}}_e = \mathbf{H}_e + \tilde{\mathbf{H}}_{e-1} = \boldsymbol{\mathcal{F}}(\tilde{\mathbf{H}}_{e-1}, \{\boldsymbol{W}_i\}) + \tilde{\mathbf{H}}_{e-1}, \tag{4}$$

where $\tilde{\mathbf{H}}_{e-1}$ and $\tilde{\mathbf{H}}_e$ respectively represent the input and output of the Riemannian residual block, $e = 3 \rightarrow$ E (when $e = 2$, $\tilde{\mathbf{H}}_{e-1}$ is replaced by $\mathbf{H}_{e-1}$ in Eqn.(4)), and $\boldsymbol{\mathcal{F}}(\tilde{\mathbf{H}}_{e-1}, \{\boldsymbol{W}_i\})$ denotes the Riemannian residual mapping. For example, $\boldsymbol{\mathcal{F}} = \boldsymbol{W}_{3_1}^T r(\boldsymbol{W}_{2_2} \tilde{\mathbf{H}}_2 \boldsymbol{W}_{2_2}^T) \boldsymbol{W}_{3_1}$ when e is set to 3, in which r signifies the ReEig operation. In what follows, another ReEig nonlinearity is applied to the generated $\tilde{\mathbf{H}}_e$ (*i.e.*, $\tilde{\mathbf{H}}_e \leftarrow r(\tilde{\mathbf{H}}_e)$). In Eqn.(4), our fundamental considerations for utilizing EWA to implement SC between SPD matrices are threefold: 1) it introduces neither parameters to be learnt nor computational complexity; 2) it can make the resulting data points still lie on the SPD manifold; 3) although the Abelian group operation (AGO) (Definition 3.1 of [2]) is faithful to the Riemannian geometry of SPD manifolds and demonstrates strong theoretical and practical benefits in Riemannian data analysis, it requires at least $\mathcal{O}(d^3)$ to achieve SC compared with EWA. Furthermore, the experimental results (reported in Section 2.1 of our supplementary material) show that although the accuracy of DreamNet-27-EWA is somewhat lower than that of DreamNet-27-AGO, its superiority in computation time is significant compared to DreamNet-27-AGO.

3.3 Objective Function

Briefly speaking, our goal is to probe a discriminative deep Riemannian network embedding to transform the input SPD matrices into more efficient and compact ones for improved classification. Taking the challenge of statistical information degradation caused by increasing the network depth into account, we establish a cascaded RAE module at the end of the backbone to reconstruct the remaining structural details from the input stage-by-stage. The built residual-like blocks facilitate the reconstruction of the remaining residual by SRAE. In addition, minimizing the reconstruction error term enables SRAE to remain highly sensitive to the variations of representations in the generated new feature manifolds, rendering the classification terms to be more effective in encoding and learning the multi-view feature distribution information. Accordingly, the loss function of the proposed method is formulated as:

$$\mathcal{L}(\theta_2, \phi; \boldsymbol{\mathcal{X}}) = \sum_{e=1}^{\mathrm{E}} \sum_{i=1}^{N} \mathcal{L}_e(\boldsymbol{X}_i, l_i) + \lambda \sum_{i=1}^{N} \mathcal{L}_2(\boldsymbol{\mathcal{Z}}_i, \hat{\mathbf{H}}_E), \tag{5}$$

where $\boldsymbol{\theta}_2 = \{\boldsymbol{\theta}_1, \boldsymbol{W}_{e_1}, \boldsymbol{W}_{e_2}, \boldsymbol{\mathcal{P}}_e\}$ ($\boldsymbol{\mathcal{P}}_e$ represents the to-be-learnt projection matrix of the FC layer of the e^{th} RAE) and λ is the trade-off parameter. In this paper, we assign a small value to λ to fine-tune the classification performance.

The first term of Eqn.(5) is the cross-entropy loss used to minimize the classification error of the input-target pairs $(\boldsymbol{X}_i, l_i)$ $(i = 1 \rightarrow N)$, implemented with the aid of the LogEig and FC layers. Specifically, $\mathcal{L}_e$ is given as:

$$\mathcal{L}_e(\boldsymbol{X}_i, l_i) = -\sum_{t=1}^{c} r(l_i, t) \times \log \frac{e^{\boldsymbol{\mathcal{P}}_e^t \boldsymbol{V}_e}}{\sum_\tau e^{\boldsymbol{\mathcal{P}}_e^\tau \boldsymbol{V}_e}}, \tag{6}$$

where $\boldsymbol{V}_e$ denotes the vectorized form of $\tilde{\mathbf{H}}_e$ ($\mathbf{H}_e$, when $e = 1$), $\boldsymbol{\mathcal{P}}_e^t$ signifies the t^{th} row of the projection matrix $\boldsymbol{\mathcal{P}}_e \in \mathbb{R}^{c \times (d_e)^2}$, and $r(l_i, t)$ is an indicator function, where $r(l_i, t) = 1$ if $l_i = t$, and 0 otherwise.

The second term of Eqn.(5) is the reconstruction error term (RT) measuring the discrepancy between the input sample and its corresponding reconstruction, computed by:

$$\mathcal{L}_2(\boldsymbol{\mathcal{Z}}_i, \hat{\mathbf{H}}_E) = ||\boldsymbol{\mathcal{Z}}_i - \hat{\mathbf{H}}_{\mathrm{E}}||_{\mathrm{F}}^2. \tag{7}$$

It is evident that the Euclidean distance (EuD) is utilized to supersede LEM for similarity measurement in Eqn.(7). Our motivations for this replacement are twofold: 1) matrix inversion can be shunned during backpropagation; 2) EuD can measure the 'statistical-level' similarity between SPD samples intuitively.

In theory, for a given pair of SPD matrices $(\boldsymbol{\mathcal{Z}}_i, \boldsymbol{\mathcal{Z}}_j)$, EuD is infinitesimal iff LEM is infinitesimal, thanks to the smoothness of matrix logarithm (Theorem 2.8 of [2]): $\forall \varepsilon, \exists \delta > 0, \forall \boldsymbol{\mathcal{Z}}_j : \|\boldsymbol{\mathcal{Z}}_i - \boldsymbol{\mathcal{Z}}_j\|_{\mathrm{F}}^2 < \delta \Rightarrow \|\log(\boldsymbol{\mathcal{Z}}_i) - \log(\boldsymbol{\mathcal{Z}}_j)\|_{\mathrm{F}}^2 < \varepsilon$. Similarly, by the smoothness of $\exp(\cdot)$ (Theorem 2.6 of [2]), the inverse map of $\log(\cdot)$, the sufficient condition of the claim mentioned above can also be proved. This theoretically indicates that the aforementioned replacement is feasible. Besides, Table 1 shows that although the use of LEM can lead to a certain improvement

Table 1. Comparison of DreamNet-27 on the AFEW dataset.

Metrics	Acc. (%)	Training time (s/epoch)
RT-EuD, *i.e.*, Eq. (7)	36.59	31.32
RT-LEM	36.71	88.16

in accuracy, the computation time required is close to three times than that of EuD, which experimentally confirms the rationality of using EuD in Eqn.(7). The experimental discussions of the role of RT are given in Section 2.2 of our supplementary material, please kindly refer to.

3.4 Motivation for Designing the SRAE Architecture

Considering that the weight matrices $\boldsymbol{W}_k$ are semi-orthogonal, *i.e.*, $\boldsymbol{W}_k^T\boldsymbol{W}_k = \boldsymbol{I}$ [23, 1, 11], inspired by the paradigm of Euclidean autoencoder, if one can design an autoencoder network with successive SPD matrix upsampling and downsampling layers in the context of SPD manifolds, its function composition would be able to asymptotically approach an identity mapping (IM) theoretically. For simplicity, we denote $\mathbf{H}_2 = \boldsymbol{W}_2^T r(\boldsymbol{W}_1\mathbf{H}_1\boldsymbol{W}_1^T)\boldsymbol{W}_2$ as the resulting SPD matrix after one upsampling and downsampling operation. As the ReEig operation only brings about minor perturbations to the eigenvalue space of the input data, under the supervision of the reconstruction term, the proposed SRAE could drive $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ close to each other, so that $||\mathbf{H}_2||_{\mathrm{F}} \rightarrow ||\mathbf{H}_1||_{\mathrm{F}}$. This design makes it possible to create an IM on the SPD manifolds, thus providing a feasible path to mitigate the degradation problem caused by increasing the network depth. In this scenario, the added shortcut connections can enable the current RAE learning phase to easily access the features of the previous stages, facilitating the reconstruction of the remaining structural details.

4 Experiments

We validate the efficacy of DreamNet[1] on three typical visual classification tasks, namely facial emotion recognition using the AFEW dataset [9], skeleton-based hand action recognition using the FPHA dataset [15], and skeleton-based human action recognition using the UAV-Human dataset [30], respectively.

4.1 Implementation

In this article, we use four layers to construct the backbone: $\boldsymbol{X}_i \rightarrow f_b^{(1)} \rightarrow f_{re}^{(2)} \rightarrow f_b^{(3)} \rightarrow f_{re}^{(4)}$, where f_b and f_{re} denote the layers of BiMap and ReEig, respectively. The stacked Riemannian autoencoder network (SRAE) is constituted by E RAEs, each of which making up five layers: $\hat{\mathbf{H}}_{e-1} \rightarrow f_b$ (input) $\rightarrow f_{re} \rightarrow f_b$ (hidden) $\rightarrow f_{re} \rightarrow f_b$ (reconstruction). Besides, the hidden layer of each RAE also connects to a classification module, consisting of three layers: $\tilde{\mathbf{H}}_e$

[1] The source code will be released on: https://github.com/GitWR/DreamNet

($\mathbf{H}_e$ when $e = 1$)$\rightarrow f_{\log} \rightarrow f_{fc} \rightarrow f_{ce}$. Wherein, $f_{\log}$, f_{fc}, and f_{ce} represent the LogEig layer, FC layer, and cross-entropy loss, respectively. In the experiments, the learning rate η is set to 0.01, the batch size B is configured as 30, and the weights of the BiMap and FC layers are initialized as random semi-orthogonal matrices and random matrices, respectively. In addition, the threshold ϵ of the ReEig layer is set to $1e$-4 for the AFEW and FPHA datasets and $1e$-5 for the UAV-Human dataset. To train our DreamNet, we use an i7-9700 (3.4GHz) PC with 16GB RAM. We found that using GPU (GTX 2080Ti) does not speed up network training. The main bottleneck seems to be the series of eigenvalue operations.

4.2 Dataset Description and Settings

AFEW Dataset: This dataset consists of 2118 video clips (split in 1741+371 fixed training and validation sets) of natural facial expressions collected from movies. For the evaluation, we follow the protocols of [23, 46] to scale down each video clip to a set of 20×20 gray-scale images, such that a 400×400 SPD matrix can be computed for video representation. On this dataset, the filter sizes of the backbone are set to 400×200 and 200×100, and those of the e^{th} RAE are configured as 100×50 and 50×100.

FPHA Dataset: This dataset includes 1,175 hand action videos belonging to 45 different categories, collected in the first-person view. For the evaluation, we follow the criterion of [15, 46] to transfer each frame into a 63-dimensional vector using the 3D coordinates of 21 hand joints provided. Hence, a total of 1,175 SPD matrices of size 63×63 can be computed, of which 600 are designated for training and the remaining 575 are used for testing. On this dataset, the filter sizes of the backbone are configured as 63×53 and 53×43, and those of the e^{th} RAE are set to 43×33 and 33×43.

UAV-Human: This dataset contains 22,476 video sequences representing 155 human action categories, collected by unmanned aerial vehicles (UAVs). Here, we first follow the practice of [6] to shape each frame (labeled by 17 major body joints with 3D coordinates) into a 51-dimensional vector. Since some actions are performed by two persons, the PCA technique is then applied to transform the 102-dimensional vectors into 51-dimensional ones, by preserving 99% energy of the data. In this case, each video can be described by an SPD matrix of size 51×51. Finally, the seventy-thirty-ratio (STR) protocol is utilized to construct the gallery and probes from the randomly picked 16,724 SPD matrices. On this dataset, the sizes of the connection weights are set to (51×43, 43×37) and (37×31, 31×37) for the backbone and the e^{th} RAE, respectively.

4.3 Ablation Studies

In this subsection, we conduct experiments to study the effectiveness of the proposed method for SPD matrix nonlinear learning.

Ablation for DreamNet: To evaluate the designed model, we carry out experiments on the AFEW, FPHA, and UAV-Human datasets to measure the

Table 2. Results on the AFEW dataset.

Networks	Acc. (%)	s/epoch	#params
DreamNet-27	36.59	31.32	0.36M
DreamNet-47	36.98	46.98	0.53M
DreamNet-92	37.47	80.62	0.95M

Table 3. Results on the FPHA dataset.

Networks	Acc. (%)	s/epoch	#params
DreamNet-27	87.78	2.60	0.11M
DreamNet-47	88.64	3.66	0.18M
DreamNet-92	88.12	6.70	0.36M

Table 4. Results on the UAV-Human dataset.

Networks	Acc. (%)	s/epoch	#params
DreamNet-27	44.88	49.04	0.10M
DreamNet-47	45.57	71.33	0.16M
DreamNet-92	46.28	129.29	0.31M

impact of the network depth on the learning capacity of the proposed model. Based on the experimental results reported in Fig. 3(a), we can make three main observations. Firstly, the inverse correlation between the depth and network accuracy is reversed with the embedding function proposed in this paper, *i.e.*, the 47-layer DreamNet (E = 5) performs better than the 27-layer DreamNet (E = 3). More importantly, the test error of DreamNet-47 is lower than that of DreamNet-27. This signifies that the degradation problem is alleviated under this design, and we succeed in improving accuracy with the increased depth. The consistency of these findings can be gleaned from Fig. 3(b) and Fig. 3(c).

Secondly, we also explore a 92-layer DreamNet by simply stacking more RAEs (E = 10 at this time). We find that compared with the 27/47-layer DreamNets, the 92-layer DreamNet achieves even lower test errors on the AFEW and UAV-Human datasets, demonstrating that the learning capacity of our network benefits from an extensive increase in the number of network layers. However, from Fig. 3(b) and Table 3, we note that the test error of DreamNet-92 is slightly higher than that of DreamNet-47 on the FPHA dataset. This could be caused by the relatively small size of this dataset. Although the benefits of depth are reflected in the classification accuracy reported in Tables 2, 3, 4, the increase in network complexity (number of parameters, #params, and training speed, s/epoch) are detrimentally affected.

Thirdly, from Fig. 3, we can see that the 27/47/92-layer DreamNets are easy to train on all the used datasets. The convergence speed of these three networks is greater than that of the original SPDNet. Note that on the AFEW dataset, the test error of our 92-layer DreamNet first shows a degradation, but eventually it recovers and exhibits performance gains. We find that this behaviour is also mirrored by the loss function on the test set. The following two factors are the main reasons for overfitting: 1) this dataset contains only 7 categories and has large intra-class diversity and inter-class ambiguity; 2) this 92-layer network may be a bit large.

Visualization: To give the reader an intuitive feeling about the proposed method in addressing the problem of structural information degradation, we choose the UAV-Human dataset as an exmaple to visualize the SPD feature maps learned by the different layers of the 27/47/92-layer DreamNets. From Fig. 4(a)-(c), we make two interesting observations: 1) for each DreamNet, compared to the low-level feature matrices, the magnitudes of the elements on the

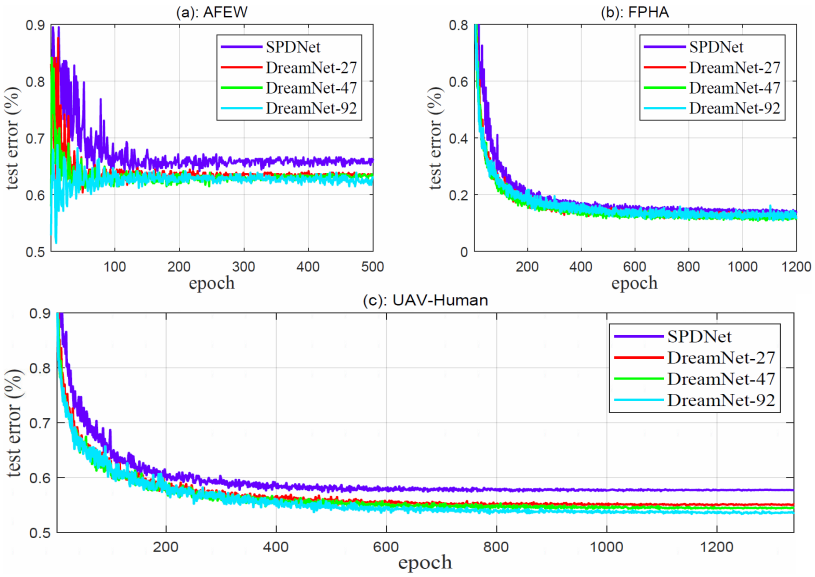


Fig. 3. The classification error of the 27/47/92-layer DreamNets versus the number of training epochs on the AFEW, FPHA, and UAV-Human datasets.

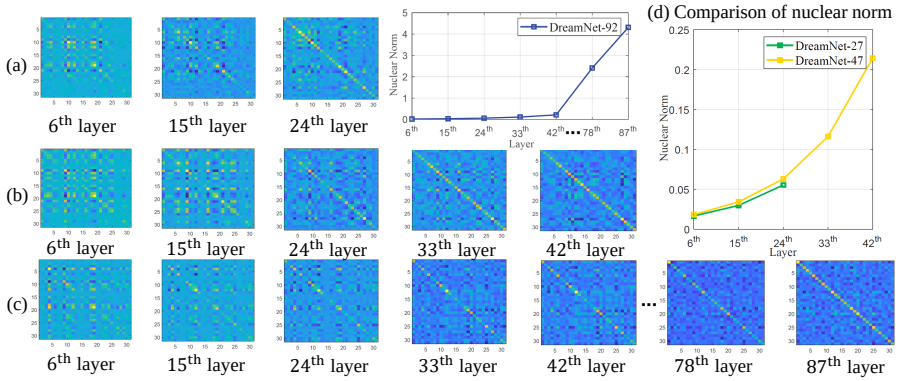
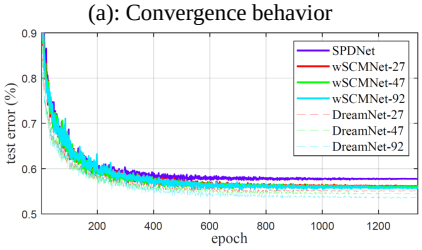


Fig. 4. The feature maps from different layers of the 27/47/92-layer DreamNets on the UAV-Human dataset are visualized in (a), (b), and (c), respectively. (d) shows the nuclear norms of these feature maps. Here, the 6^{th} layer is actually the hidden layer of the first RAE, and the other layers are actually used to realize element-wise addition.

main diagonal of the high-level feature matrices are becoming larger, while the off diagonal ones are getting smaller; 2) with increasing the network depth, this concentration of energy becomes even more significant. Besides, the nuclear norms shown in Fig. 4(d) reflect that the deeper the learned features, the lower their redundancy. These results suggest that the continuous incremental learning on the remaining residuals can enable the proposed network to capture pivotal structural information embodied in the original data points, thus being helpful for classification.

(a): Convergence behavior



(b): Classification accuracy

Methods	Acc. (%)
wSCMNet-27	43.86
wSCMNet-47	44.04
wSCMNet-92	44.40

Fig. 5. Performance on the UAV-Human dataset

Ablation study for the Shortcut Connections: To verify the benefits of the shortcut connections (SCs), we make experiments to study the performance of a simplified DreamNet (named wSCMNet) obtained by removing the SCs from SRAE module. We choose the UAV-Human dataset as an example. It can be seen from Fig. 5(a) that the wSCMNet with different depths can converge to a better solution in less than 1,300 epochs, indicating that it has a good convergence behavior. However, the classification scores of 27/47/92-layer wSCMNets tabulated in Fig. 5(b) are lower than those of 27/47/92-layer DreamNets. In spite of this, they are still better than those of the competitors listed in Table 5. From Fig. 5(a), we also find that the convergence speed of DreamNets is slightly faster than that of wSCMNets. These experimental results not only demonstrate the effectiveness of the proposed SRAE network, but also confirm that the SCs can: 1) enhance the representational capacity of SRAE module; 2) simplify the training of deeper networks. The underlying reason is that this operation facilitates the information interaction between different RAEs.

Ablation of the Classification Module: In this part, we make experiments on the FPHA dataset as an example to investigate the impact of the number of classification modules on the accuracy of DreamNet (here we take DreamNet-27 as an example) in the test phase. From Fig. 6(a), we can see that: 1) the greater the number of classifiers, the higher the accuracy; 2) the 3^{rd} classifier are more effective than the others. This not only indicates that these classifiers are complementary to each other, but also demonstrates that the higher-level features are more informative.

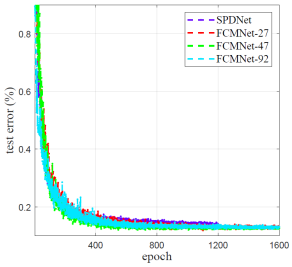
Inspired by this experiment, we then investigated how the performance of DreamNet is affected by removing the first E-1 classification modules from SRAE (we name the simplified DreamNet FCMNet here). In this case, we find that the initial learning rate of 0.01 is a bit too small for the 47/92-layer FCMNets. So we respectively assign the initial learning rates of 0.02 and 0.05 to FCMNet-47 and FCMNet-92, and make them attenuate by a factor of 0.9 every 100 epochs. It is evident that the studied FCMNets converge well (Fig. 6(b)). Although the accuracy of 27/47/92-layer FCMNets (87.18%, 87.60%, and 87.30%) are somewhat inferior to that of 27/47/92-layer DreamNets, they are still better than those of the competitors listed in Table 6. These observations again certify the effectiveness of our design in overcoming the degradation problem and learning a powerful manifold-to-manifold deep transformation mapping. Besides, Fig. 6(c) not only further indicates that the residual mapping $\boldsymbol{\mathcal{F}}$ is not close to a zero

(a): Combination of different CLAs

Methods	Acc. (%)
1st CLA	83.48
2nd CLA	85.78
3rd CLA	87.39
1st &2nd CLAs	85.91
1st &3rd CLAs	87.70
2nd &3rd CLAs	87.70
All the CLAs	**87.78**

CLA: Classifier

(b): Convergence behavior



(c) Visualization. Left: layers of FCMNet-92; Right: layers of DreamNet-92

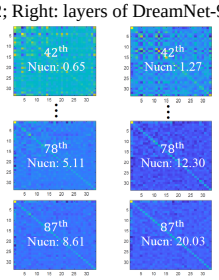


Fig. 6. Performance on the FPHA dataset, where 'Nucn' represents the nuclear norm.

Table 5. Accuracy (%) on the AFEW and UAV-Human datasets.

Methods	AFEW	UAV-Human
GDA [16]	29.11	28.13
CDL [48]	31.81	31.11
PML [25]	28.98	10.66
HERML [24]	32.14	34.18
HRGEML [6]	35.89	36.10
SPDML[19]	26.72	22.69
GEMKML [45]	35.71	34.67
DeepO2P [28]	28.54	N/A
DARTS [31]	25.87	36.13
FairDARTS [7]	25.34	40.01
GrNet [27]	34.23	35.23
SPDNet [23]	34.23	42.31
SPDNetBN [4]	36.12	43.28
ManifoldNet [5]	23.98	N/A
SymNet [46]	32.70	35.89
DreamNet-27	36.59	44.88
DreamNet-47	36.98	45.57
DreamNet-92	**37.47**	**46.28**

Table 6. Accuracy (%) on the FPHA dataset.

Methods	Color	Depth	Pose	Acc.
Two streams [12]	✓	✗	✗	75.30
Novel View [35]	✗	✓	✗	69.21
Lie Group [42]	✗	✗	✓	82.69
HBRNN [10]	✗	✗	✓	77.40
LSTM [15]	✗	✗	✓	80.14
JOULE[22]	✓	✓	✓	78.78
Gram Matrix [52]	✗	✗	✓	85.39
TF [14]	✗	✗	✓	80.69
TCN [29]	✗	✗	✓	78.57
ST-GCN [50]	✗	✗	✓	81.30
H+O [39]	✓	✗	✗	82.43
TTN [32]	✗	✗	✓	83.10
DARTS [31]	✗	✗	✓	74.26
FairDARTS [7]	✗	✗	✓	76.87
SPDML[25]	✗	✗	✓	76.52
HRGEML [6]	✗	✗	✓	85.04
SPDNet [23]	✗	✗	✓	86.26
SPDNetBN [4]	✗	✗	✓	86.83
SymNet [46]	✗	✗	✓	82.96
DreamNet-27	✗	✗	✓	87.78
DreamNet-47	✗	✗	✓	**88.64**
DreamNet-92	✗	✗	✓	88.12

mapping, but also shows that the multi-classifier learning (MCL) scheme of DreamNet can produce more efficient deep features with lower redundancy. Since the use of multiple classifiers can provide sufficient supervision information, and the increase in training time is slight (*e.g.*, one training epoch lasted on average 4.51s for FCMNet-92, and 6.70s for DreamNet-92 on this dataset), we adopt the MCL mechanism in this article.

4.4 Comparison with State-of-the-art Methods

For a fair comparison, based on the publicy available source codes, we follow the original recommendations to tune the parameters of each comparative method,

and report their best results on all three datasets. For DARTS and FairDARTS, we run their official implementations with default settings in the SPD matrix logarithmic domain. For DeepO2P, its classification accuracy on the AFEW dataset is provided by [23]. Since ManifoldNet requires SPD data points with multiple channels, it is inapplicable to the FPHA and UAV-Human skeleton datasets. From Table 5, it is evident that our 27-layer DreamNet outperforms all the involved competitors on the AFEW and UAV-Human datasets. Besides, with the network depth (the number E of the cascaded RAEs) increases, the accuracy of the 47/92-layer DreamNets is monotonically improving. Here, we also select some popular action recognition methods for better comparison on the FPHA dataset. Table 6 shows that our 27/47/92-layer DreamNets are the best performers for the hand action recognition task. For further evaluation, an aggressively deep model of over 180 layers has also been explored on the UAV-Human dataset. We set E = 20 that leads to a 182-layer DreamNet. The experimental results (reported in Section 2.3 of our supplementary material) show that it has no difficulty in optimization, and the classification accuracy (46.03%) achieved is still fairly good. These observations confirm that the suggested deep learning mechanism over the original SPD network is effective for improving the visual classification performance.

5 Conclusion

In this paper, we proposed an effective methodology for increasing the depth of SPD neural networks without destroying the geometric information conveyed by the input data. This is achieved by proposing a novel cascading network architecture with multiple Riemannian autoencoder learning stages appended to the backbone SPD network to enrich the deep layers of structured representations. Thanks to the insertion of innovative residual-like blocks via shortcut connections, a better incremental learning of residual structural details can be facilitated. The experimental results suggest that our Riemannian network is an effective solution against the geometric information degradation problem, with favourable performance compared to the state-of-the-art methods. For future work, we plan to develop an adaptive criterion that would enable an automatic assessment of the relative significance of the generated feature maps. This would facilitate the use of a neural architecture search (NAS) technique to adapt the proposed network to different pattern recognition tasks.

Acknowledgements This work was supported by the National Natural Science Foundation of China (62020106012, U1836218, 61672265, 62106089, 62006097), the 111 Project of Ministry of Education of China (B12018), the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX21-2006), and the UK EPSRC EP/N007743/1, MURI/EPSRC/DSTL EP/R018456/1 grants.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press (2009)

2. Arsigny, V., Fillard, P., Pennec, X., Ayache, N.: Geometric means in a novel vector space structure on symmetric positive-definite matrices. SIAM J. Matrix Anal. Appl. pp. 328–347 (2007)
3. Barachant, A., Bonnet, S., Congedo, M., Jutten, C.: Classification of covariance matrices using a riemannian-based kernel for bci applications. Neurocomputing pp. 172–178 (2013)
4. Brooks, D., Schwander, O., Barbaresco, F., Schneider, J.Y., Cord, M.: Riemannian batch normalization for spd neural networks. arXiv preprint arXiv:1909.02414 (2019)
5. Chakraborty, R., Bouza, J., Manton, J., Vemuri, B.C.: Manifoldnet: A deep neural network for manifold-valued data with applications. IEEE Trans. Pattern Anal. Mach. Intell. pp. 799–810 (2022)
6. Chen, Z., Xu, T., Wu, X.J., Wang, R., Kittler, J.: Hybrid riemannian graph-embedding metric learning for image set classification. IEEE Trans. Big Data (doi: 101109/TBDATA20213113084, 2021)
7. Chu, X., Zhou, T., Zhang, B., Li, J.: Fair darts: Eliminating unfair advantages in differentiable architecture search. In: ECCV pp. 465–480 (2020)
8. Dai, M., Zhang, Z., Srivastava, A.: Analyzing dynamical brain functional connectivity as trajectories on space of covariance matrices. IEEE Trans. Med. Imaging pp. 611–620 (2019)
9. Dhall, A., Goecke, R., Joshi, J., Sikka, K., Gedeon, T.: Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In: ICMI pp. 461–466 (2014)
10. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: CVPR pp. 1110–1118 (2015)
11. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. pp. 303–353 (1998)
12. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: CVPR pp. 1933–1941 (2016)
13. Gao, Z., Wu, Y., Harandi, M., Jia, Y.: A robust distance measure for similarity-based classification on the spd manifold. IEEE Trans. Neural Netw. Learn. Syst. pp. 3230–3244 (2020)
14. Garcia-Hernando, G., Kim, T.K.: Transition forests: Learning discriminative temporal transitions for action recognition and detection. In: CVPR pp. 432–440 (2017)
15. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.K.: First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In: CVPR pp. 409–419 (2018)
16. Hamm, J., Lee, D.D.: Grassmann discriminant analysis: a unifying view on subspace-based learning. In: ICML pp. 376–383 (2008)
17. Harandi, M., Salzmann, M.: Riemannian coding and dictionary learning: Kernels to the rescue. In: CVPR pp. 3926–3935 (2015)
18. Harandi, M., Salzmann, M., Hartley, R.: Joint dimensionality reduction and metric learning: A geometric take. In: ICML pp. 1404–1413 (2017)
19. Harandi, M., Salzmann, M., Hartley, R.: Dimensionality reduction on spd manifolds: The emergence of geometry-aware methods. IEEE Trans. Pattern Anal. Mach. Intell. pp. 48–62 (2018)
20. Harandi, M., Sanderson, C., Hartley, R., Lovell, B.C.: Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In: ECCV pp. 216–229 (2012)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR pp. 770–778 (2016)

22. Hu, J.F., Zheng, W.S., Lai, J., Zhang, J.: Jointly learning heterogeneous features for rgb-d activity recognition. In: CVPR pp. 5344–5352 (2015)
23. Huang, Z., Van Gool, L.: A riemannian network for spd matrix learning. In: AAAI pp. 2036–2042 (2017)
24. Huang, Z., Wang, R., Shan, S., Chen, X.: Hybrid euclidean-and-riemannian metric learning for image set classification. In: ACCV pp. 562–577 (2014)
25. Huang, Z., Wang, R., Shan, S., Chen, X.: Projection metric learning on grassmann manifold with application to video based face recognition. In: CVPR pp. 140–149 (2015)
26. Huang, Z., Wang, R., Shan, S., Li, X., Chen, X.: Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In: ICML pp. 720–729 (2015)
27. Huang, Z., Wu, J., Van Gool, L.: Building deep networks on grassmann manifolds. In: AAAI pp. 1137–1145 (2018)
28. Ionescu, C., Vantzos, O., Sminchisescu, C.: Training deep networks with structured layers by matrix backpropagation. arXiv preprint arXiv:1509.07838 (2015)
29. Kim, T.S., Reiter, A.: Interpretable 3d human action analysis with temporal convolutional networks. In: CVPRW pp. 1623–1631 (2017)
30. Li, T., Liu, J., Zhang, W., Ni, Y., Wang, W., Li, Z.: Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles. In: CVPR pp. 16266–16275 (2021)
31. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. In: ICLR (2019)
32. Lohit, S., Wang, Q., Turaga, P.: Temporal transformer networks: Joint learning of invariant and discriminative time warping. In: CVPR pp. 12426–12435 (2019)
33. Nguyen, X.S., Brun, L., Lézoray, O., Bougleux, S.: A neural network based on spd manifold learning for skeleton-based hand gesture recognition. In: CVPR pp. 12036–12045 (2019)
34. Pennec, X., Fillard, P., Ayache, N.: A riemannian framework for tensor computing. Int. J. Comput. Vis. pp. 41–66 (2006)
35. Rahmani, H., Mian, A.: 3d action recognition from novel viewpoints. In: CVPR pp. 1506–1515 (2016)
36. Sanin, A., Sanderson, C., Harandi, M.T., Lovell, B.C.: Spatio-temporal covariance descriptors for action and gesture recognition. In: WACV Workshop pp. 103–110 (2013)
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
38. Sun, H., Zhen, X., Zheng, Y., Yang, G., Yin, Y., Li, S.: Learning deep match kernels for image-set classification. In: CVPR pp. 3307–3316 (2017)
39. Tekin, B., Bogo, F., Pollefeys, M.: H+o: Unified egocentric recognition of 3d hand-object poses and interactions. In: CVPR pp. 4511–4520 (2019)
40. Tosato, D., Farenzena, M., Spera, M., Murino, V., Cristani, M.: Multi-class classification on riemannian manifolds for video surveillance. In: ECCV pp. 378–391 (2010)
41. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE Trans. Pattern Anal. Mach. Intell. pp. 1713–1727
42. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: CVPR pp. 588–595 (2014)
43. Vemulapalli, R., Pillai, J.K., Chellappa, R.: Kernel learning for extrinsic classification of manifold features. In: CVPR pp. 1782–1789 (2013)

44. Wang, R., Wu, X.J., Chen, Z., Xu, T., Kittler, J.: Learning a discriminative spd manifold neural network for image set classification. Neural Netw. pp. 94–110 (2022)
45. Wang, R., Wu, X.J., Kittler, J.: Graph embedding multi-kernel metric learning for image set classification with grassmann manifold-valued features. IEEE Trans. Multimedia pp. 228–242 (2021)
46. Wang, R., Wu, X.J., Kittler, J.: Symnet: A simple symmetric positive definite manifold deep learning method for image set classification. IEEE Trans. Neural Netw. Learn. Syst. pp. 2208–2222 (2022)
47. Wang, R., Wu, X.J., Xu, T., Hu, C., Kittler, J.: Deep metric learning on the spd manifold for image set classification. IEEE Trans. Circuits Syst. Video Technol. (2022)
48. Wang, R., Guo, H., Davis, L.S., Dai, Q.: Covariance discriminative learning: A natural and efficient approach to image set classification. In: CVPR pp. 2496–2503 (2012)
49. Xu, T., Feng, Z.H., Wu, X.J., Kittler, J.: An accelerated correlation filter tracker. Pattern Recognit. pp. 1–10 (2020)
50. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI (2018)
51. Zhang, T., Zheng, W., Cui, Z., Zong, Y., Li, C., Zhou, X., Yang, J.: Deep manifold-to-manifold transforming network for skeleton-based action recognition. IEEE Trans. Multimedia pp. 2926–2937 (2020)
52. Zhang, X., Wang, Y., Gou, M., Sznaier, M., Camps, O.: Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In: CVPR pp. 4498–4507 (2016)
53. Zhao, S., Xu, T., Wu, X.J., Zhu, X.F.: Adaptive feature fusion for visual object tracking. Pattern Recognit. pp. 1–11 (2021)
54. Zhou, L., Wang, L., Zhang, J., Shi, Y., Gao, Y.: Revisiting metric learning for spd matrix based visual representation. In: CVPR pp. 3241–3249 (2017)
55. Zhu, X.F., Wu, X.J., Xu, T., Feng, Z.H., Kittler, J.: Complementary discriminative correlation filters based on collaborative representation for visual object tracking. IEEE Trans. Circuits Syst. Video Technol. pp. 557–568 (2020)