Riemannian Local Mechanism for SPD Neural Networks

Ziheng Chen[1], **Tianyang Xu**[1], **Xiao-Jun Wu**[*1], **Rui Wang**[1], **Zhiwu Huang**[2], **Josef Kittler**[3]

[1]School of Artificial Intelligence and Computer Science, Jiangnan University
[2]School of Computing and Information Systems, Singapore Management University
[3]Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey
ziheng_ch@163.com, {tianyang.xu, wu_xiaojun,cs_wr}@jiangnan.edu.cn, zwhuang@smu.edu.sg, j.kittler@surrey.ac.uk

Abstract

The Symmetric Positive Definite (SPD) matrices have received wide attention for data representation in many scientific areas. Although there are many different attempts to develop effective deep architectures for data processing on the Riemannian manifold of SPD matrices, very few solutions explicitly mine the local geometrical information in deep SPD feature representations. Given the great success of local mechanisms in Euclidean methods, we argue that it is of utmost importance to ensure the preservation of local geometric information in the SPD networks. We first analyse the convolution operator commonly used for capturing local information in Euclidean deep networks from the perspective of a higher level of abstraction afforded by category theory. Based on this analysis, we define the local information in the SPD manifold and design a multi-scale submanifold block for mining local geometry. Experiments involving multiple visual tasks validate the effectiveness of our approach. The supplement and source code can be found in https://github.com/GitZH-Chen/MSNet.git.

1 Introduction

Symmetric Positive Definite (SPD) matrices have shown great success in many scientific areas, like medical image analysis(Chakraborty et al. 2020, 2018), elasticity (Guilleminot and Soize 2012), signal processing (Hua et al. 2017; Brooks et al. 2019), machine learning (Kulis, Sustik, and Dhillon 2006; Harandi et al. 2012), and computer vision (Chakraborty 2020; Zhang et al. 2020; Zhen et al. 2019; Huang and Van Gool 2017; Nguyen 2021; Harandi, Salzmann, and Hartley 2018). However, the non-Euclidean nature of SPD matrices precludes the use of a wide range of data analysis tools, the applicability of which is confined to Euclidean metric spaces. Motivated by this problem, a number of Riemannian metrics have been introduced, such as Log-Euclidean metric (LEM) (Arsigny et al. 2005), Affine-Invariant Riemannian metric (AIM) (Pennec, Fillard, and Ayache 2006), Log-Cholesky metric (LCM) (Lin 2019).

With these well-studied Riemannian metrics, some Euclidean techniques can be generalized into SPD manifolds. Some approaches adopted approximation methods that locally flatten the manifold by identifying it with its tangent space(Huang et al. 2015c), or by projecting the manifold into a Reproducing Kernel Hilbert Spaces (RKHS)(Chen et al. 2021; Harandi et al. 2012; Wang et al. 2012). However, these methods tend to distort the geometrical structure communicated by the data. To address this issue, (Harandi, Salzmann, and Hartley 2018; Gao et al. 2019b) proposed direct learning algorithms on SPD manifolds. In addition, inspired by the significant progress achieved by deep learning (LeCun et al. 1998; Krizhevsky, Sutskever, and Hinton 2012), (Huang and Van Gool 2017; Zhang et al. 2020; Brooks et al. 2019; Chakraborty et al. 2018, 2020; Chakraborty 2020; Nguyen 2021; Zhen et al. 2019; Wang et al. 2022a,b) attempted to build deep Riemannian networks for non-linear representation learning on SPD matrices.

Nevertheless, many existing deep SPD networks treat the SPD features as a global representation. Given the success of multi-scale features in both conventional feature design (Belongie, Malik, and Puzicha 2002; Lowe 2004) and deep learning (Szegedy et al. 2015; Krizhevsky, Sutskever, and Hinton 2012), it should be rewarding to investigate local mechanism in Riemannian neural networks. Accordingly, in this paper, we develop a deep multi-scale submanifold network designed to capture the informative local geometry in deep SPD networks. To the best of our knowledge, this is the first work to successfully mine the local geometric information on SPD manifolds.

As convolution is one of the most successful techniques for dealing with local information in traditional deep learning, we first analyze its mathematical essence from the perspective of category theory, to identify the universal property which is transferable to manifolds. We proceed to define the local information in the category of SPD manifolds and propose multi-scale submanifold blocks to capture both holistic and local geometric information. In summary, our contributions are three-fold: 1). a theoretical guideline is developed for the Riemannian local mechanism. 2). local patterns in Riemannian manifolds are rigorously defined. 3). a novel multi-scale submanifold block is proposed to capture vibrant local statistical information on the SPD networks.

2 Related Work

To take advantage of deep learning techniques, some effort has been made to generalize Euclidean deep learning into a Riemannian one. (Huang and Van Gool 2017) de-

[*]Corresponding author.

signed a densely connected feedforward network on SPD matrices with a procedure involving a slice of spectral layers. (Chakraborty et al. 2020) proposed a theoretical framework to fulfil convolution network on Riemannian manifolds, where each 'pixel' of the input tensor is required to be a manifold-valued point. However, different from Euclidean convolution, none of these methods pay attention to the local information in a single SPD matrix. In contrast, (Zhang et al. 2020) proposed an SPD transformation network for action recognition. They designed an SPD convolutional layer, which is similar to the Euclidean convolution except that the convolutional kernels are required to be SPD. Note that the square matrices covered by a sliding window might not be SPD. Therefore, local geometry might be distorted or omitted, undermining the efficacy of their proposed network. In contrast, in our approach, the proposed mechanisms can faithfully preserve local information. A multi-scale representation is further adopted, which captures different levels of statistical information. We expect that Riemannian networks can benefit from this comprehensive statistical information.

3 Preliminaries

To develop our proposed method, category theory and regular submanifolds are briefly reviewed. Due to the page limit, others such as differential manifolds, the geometry of SPD manifolds, and our backbone network, SPDNet, are presented in the supplement.

3.1 Foundations of Category Theory

Category theory, which is similar to object-oriented programming in computer science, studies the universal properties and mathematical abstractions shared by different domains.

Definition 1. A category $\mathcal{C}$ consists of a collection of elements, called objects, denoted by $\mathrm{Obj}(\mathcal{C})$, and a set $\mathrm{Mor}(A,B)$ of elements, called morphisms from A to B, for any two objects $A, B \in \mathrm{Obj}(\mathcal{C})$. Morphisms should satisfy the below three axioms:

- composition: given any $f \in \mathrm{Mor}(A,B)$ and $g \in \mathrm{Mor}(B,C)$, the composition $h = g \circ f \in \mathrm{Mor}(A,C)$ is well-defined.
- identity: for each object A, there is an identity morphism $1_A \in \mathrm{Mor}(A,A)$ such that for any $f \in \mathrm{Mor}(A,B)$ and $g \in \mathrm{Mor}(B,A)$,
$$f \circ 1_A = f, 1_A \circ g = g; \tag{1}$$
- associative: for $f \in \mathrm{Mor}(A,B), g \in \mathrm{Mor}(B,C)$, and $h \in \mathrm{Mor}(C,D)$,
$$h \circ (g \circ f) = (h \circ g) \circ f. \tag{2}$$

The set of all the morphisms in $\mathcal{C}$ is denoted as $\mathrm{Mor}(\mathcal{C})$.

Let us take the linear space, which is widely studied in pattern recognition, as an example. In this category, the objects are linear spaces and morphisms are linear homomorphisms. More details are introduced in (Tu 2011, § 10).

Category theory enables us to develop a mathematical abstraction of operations in one category and generalize them into another category. In this paper, we will rely on this theory to derive local mechanisms in Riemannian manifolds from Euclidean counterparts.

3.2 Regular Submanifolds

Regular submanifolds (Tu 2011, § 9) of manifolds generalize the idea of subspace in the Euclidean space. In the smooth category, since manifolds are locally Euclidean, submanifolds are defined locally.

Definition 2. A subset $\mathcal{S}$ of a smooth manifold $\mathcal{N}$ of dimension n is a regular submanifold of dimension k if for every $p \in \mathcal{S}$ there is a coordinate neighbourhood $(U, \phi) = \left(U, x^1, \ldots, x^n\right)$ of p in the maximal atlas of $\mathcal{N}$ such that $U \cap \mathcal{S}$ is defined by the vanishing of $n-k$ of the coordinate functions.

4 Proposed Method

In this section, we will introduce our method in detail. As Euclidean convolution is one of the most successful local mechanisms, we first analyze this operation from the perspective of category theory to uncover the mathematical essence of the Euclidean local mechanism. Under this analysis, we proceed to define the local pattern in the SPD manifold. Finally, we introduce a multi-scale local mechanism to capture fine-grained local information. The proposed network is conceptually illustrated in Figure 1.

4.1 Analysis of Euclidean Convolution

Since the convolution is an operation in the category of linear space, we first consider it from the perspective of linear algebra. Then we will proceed with the analysis via category theory to derive the general properties that can be transferred into the SPD manifold.

To avoid tedious subscripts, we consider an example whose input and output are single channel tensors. In this case, only one kernel filter is involved, but the following analysis can also be readily generalized into an arbitrary number of channels. In convolution networks, the tensor feature and the i^{th} receptive field can be viewed as the linear space V and subspace V_i, respectively. The process that the i^{th} receptive field is reduced into a real scalar by a specific kernel filter can be deemed as a linear mapping $f_i : V_i \rightarrow M_i$. Note that M_i is a trivial one-dimensional linear space, $\mathbb{R}$. After convolution, each receptive field is reduced into a real number and these scalar elements are concatenated to build a new tensor feature. This process can be more generally described by the notion of direct sum "$\oplus$" (Roman, Axler, and Gehring 2005) in linear algebra, *i.e.,* $M = M_1 \oplus \cdots \oplus M_n$. Not that the direct sum, "$\oplus$" can be intuitively understood as a counterpart of the union in the set theory. (see supplement for more details) The above analysis leads to the following abstraction of the convolution operation.

Proposition 1. *For a given linear space V of dimension $d \times d$, n linear subspaces $V_1, V_2, \cdots, V_n$ of dimension $k \times k$*

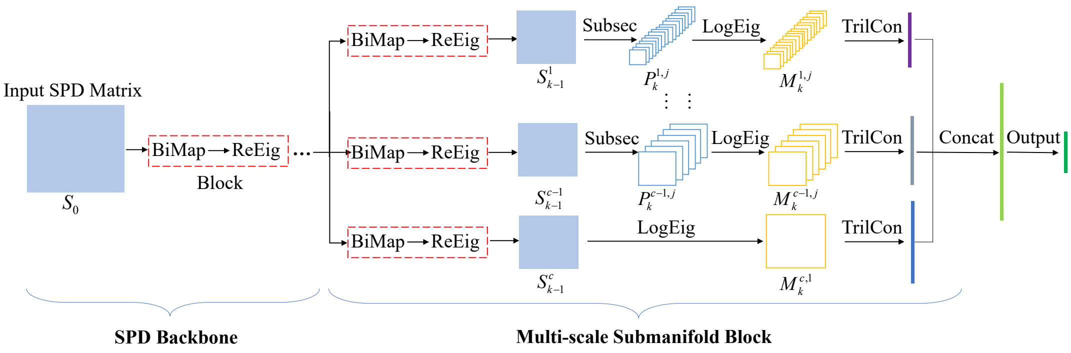

Figure 1: Illustration of the proposed Multi-scale Submanifold Network (MSNet). We first employ SPDNet (Huang and Van Gool 2017) as our backbone to extract lower dimensional, yet more discriminative SPD feature representations. Then in each branch, a BiMap-ReEig block is exploited to obtain SPD representations S_{k-1}^i, where $k-1$ and i are layer and channel index respectively. We propose a submatrix selection, denoted as SubSec, to result in $P_k^{i,1} \cdots P_k^{i,n^i}$ along the i^{th} channel for local manifold feature learning, where n^i is the number of selected submanifolds in the i^{th} channel. Next, LogEig layer is applied to map each submatrix feature into a Euclidean space, *i.e.,* $M_k^{i,j} = \log(P_k^{i,j})$. Then, we apply TrilCan to go through a process of extracting a lower triangular matrix, vectorization, and concatenation. Finally, we concatenate all the vectors from the different branches with a Concat layer, followed by any regular output layers like the softmax layer. Note that the SPD matrix itself can also be viewed as a trivial submanifold, encoding global information, and hence the bottom branch is exploited to capture global information, which is how SPDNet works.

are selected and a linear function $f_i(\cdot) : V_i \to M_i$ is performed in each of them to extract local linear information. The resulting linear spaces $M_1, \cdots, M_n$ are combined into a final linear space M by direct sum, i.e.,$M = M_1 \oplus \cdots \oplus M_n$.

To discover a more general property of Euclidean convolution, we further analyze it by category theory. To this end, we can simply substitute the linear algebra terms with category language following the axioms of category theory. In detail, linear space V, subspace V_i and linear function f_i can be more generally described as object A, sub-object A_i and morphism f_i. Besides, we notice that each subspace V_i shares the same dimensionality, which indicates $V_1, \cdots, V_n$ are equivalent in the sense of linear space. This suggests that the extracted sub-objects $A_1, \cdots, A_n$ should be isomorphic. However, not all categories share the idea of the direct sum. For example, the categories, known as group, ring and field, do not have this kind of operation. Therefore, the combination strategies vary in different categories. Now, we could obtain a more general description of convolution by category theory.

Proposition 2. *In a category $\mathcal{C}$, for an object $A \in Obj(\mathcal{C})$, we extract n isomorphic sub-objects from A, denoted by $A_1, A_2, \ldots, A_n$. Then morphism $f_i \in \operatorname{Hom}(A_i, B_i)$ is applied to each sub-object to map it into a resulting object B_i. The resulting objects $B_1, \cdots, B_i$ are combined into a final object $B \in Obj(\mathcal{C})$ according to certain principles.*

With the Proposition 2 as an intermediary, we can generalize the convolution into manifolds theoretically. Specifically, the object, sub-object and morphism in manifolds are manifolds, submanifolds and smooth maps respectively.

Proposition 3. *For a manifold $\mathcal{M}$, we extract n isomorphic submanifolds $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_n$ and map each one of them by a smooth maps $f_i(\cdot) : \mathcal{M}_i \to \mathcal{M}_i'$. The resulting manifolds $\mathcal{M}_i'$ are aggregated into a final manifold $\mathcal{M}'$ according to certain principles.*

As a summary of the above discussion, we obtain the following important insights about the Riemannian local mechanism. First of all, the local patterns in manifolds are submanifolds. Secondly, not all submanifolds are involved, and all the selected submanifolds could be isomorphic. Lastly, a specific way of aggregating submanifolds should be elaborately designed according to the axioms of manifolds.

4.2 Submanifolds in SPD Manifolds

Proposition 3 demonstrates that local patterns in manifolds are submanifolds. In this subsection, we will identify the submanifolds in the specific SPD manifolds. Briefly speaking, in the category of SPD manifolds, submanifolds should further be SPD manifolds. This constraint can be fulfilled by principle submatrices.

For a clearer description, let us make some notifications first. Denote k-fold row and column indices as $\mathcal{I} = \{i_1, \cdots, i_k\}$ and $\mathcal{J} = \{j_1, \cdots, j_k\}$. For a set of real square matrices $\mathbb{R}^{n\times n}$, we denote $(\mathbb{R}^{n\times n})_{\mathcal{I},\mathcal{J}}$ as the set of submatrices, which are obtained by remaining rows $\mathcal{I}$ and columns $\mathcal{J}$. If $\mathcal{I} = \mathcal{J}$, then $(\mathbb{R}^{n\times n})_{\mathcal{I},\mathcal{I}}$ is the set of principal submatrices, abbreviated as $(\mathbb{R}^{n\times n})_{\mathcal{I}}$.

As we discussed before, subspaces are sub-objects in the category of linear algebra. For a set of real square matrices $\mathbb{R}^{n\times n}$, any set of $k \times k$ submatrices, $(\mathbb{R}^{n\times n})_{\mathcal{I},\mathcal{J}}$, forms a subspace of $\mathbb{R}^{n\times n}$. However, things would be different for SPD manifolds. Linear algebra tells us that an arbitrary submatrix of an SPD matrix might not be SPD, and even not be

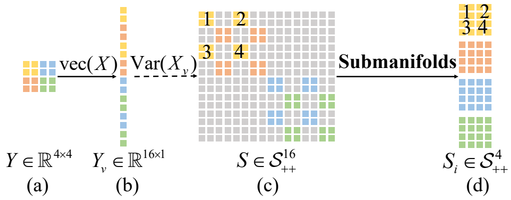
vec(X)
Var(X_v)
Submanifolds
1
2
3
4
1 2
3 4
$Y \in \mathbb{R}^{4\times4}$
(a)
$Y_v \in \mathbb{R}^{16\times1}$
(b)
$S \in \mathcal{S}_{++}^{16}$
(c)
$S_i \in \mathcal{S}_{++}^{4}$
(d)

Figure 2: Illustration of the process of selecting principal submatrices. (a) We deem an $S \in \mathcal{S}_{++}^{16}$ as a covariance from an imaginary 4×4 random matrix Y. We use a 2×2 sliding window with skip of 2 on X to obtain the corresponding position index. We use four kinds of colour to denote the four regions of interest. (b) shows the corresponding indexes of the four regions in vectorized Y_v. (c) Then we can find the corresponding region covariance matrices for the four regions from S. (d) The region covariance matrices $S_i \in \mathcal{S}_{++}^{4}$ corresponding to these local regions are the submanifolds we select.

symmetric, while principle submatrices are always SPD. Although we can readily prove that $(\mathcal{S}_{++}^{n})_{\mathcal{I},\mathcal{J}}$ can be viewed as a regular submanifold of the SPD manifold $\mathcal{S}_{++}^{n}$, $(\mathcal{S}_{++}^{n})_{\mathcal{I},\mathcal{J}}$ might not be a SPD manifold. This could cause some inconsistency, since in the specific category of SPD manifolds, objects should always be SPD manifolds. In addition, $(\mathcal{S}_{++}^{n})_{\mathcal{I}}$ can be viewed as a regular submanifold of $\mathcal{S}_{++}^{n}$. The above discussion is formalized by the following theorem. (Proof is presented in the supplement.)

Theorem 1. *For an SPD manifold $\mathcal{S}_{++}^{n}$, the set of principal submatrices $(\mathcal{S}_{++}^{n})_{\mathcal{I}}$ is an SPD manifold and can be embedded into the $\mathcal{S}_{++}^{n}$ as a regular submanifold. In addition, for any proper indices $\mathcal{I}$ and $\mathcal{J}$ satisfying $|\mathcal{I}| = |\mathcal{J}|$, $(\mathcal{S}_{++}^{n})_{\mathcal{I}}$ is isomorphic to $(\mathcal{S}_{++}^{n})_{\mathcal{J}}$.*

Now, we have identified that the local pattern in the specific SPD manifolds are principal submatrices.

4.3 Multi-scale Submanifold Block

Because of the analysis in Section 4.1, there are two factors we should consider when designing our submanifold block, *i.e.*, the rule for selecting isomorphic submanifolds and the way of aggregating them. In the following, we first discuss the details of selecting submanifolds in the SPD manifold. Then we proceed to introduce our multi-scale submanifold block, which fuses submanifolds via a divide-aggregation strategy.

The Principles for Selecting Submanifolds Theorem 1 reveals that the isomorphic submanifolds in SPD manifolds are the sets of principal submatrices of the same size. Note that incorporating all the sub-objects might not be an optimal solution, since it introduces redundant information with cumbersome computation. In conventional convolution, sub-objects are selected following the concept of the receptive field. In terms of specific SPD manifolds, the number of principal submatrices grows combinatorially, which could cause a dimensionality explosion. For instance, if we select all the principal submatrices of 4×4 from a 16×16 SPD matrix, which seems like a trivial example, the total number of selected submatrices would be $C_{16}^{4} = 1820$. In most cases, this huge number of small SPD submatrices would not be manageable. Therefore, it is crucial to select principal submatrices to construct submanifolds. To this end, we introduce a method for selecting principal submatrices.

In probability theory, the distribution information of a $d \times d$ random matrix X can be conveyed by its covariance matrix S of $d^2 \times d^2$, where $S = \mathrm{Var}(\mathrm{vec}(X))$ and $\mathrm{vec}(\cdot)$ denotes vectorization. The covariance of a $k \times k$ local region in X corresponds to a $k^2 \times k^2$ principal submatrix of S. If we focus on multiple $k \times k$ local regions, we then can extract a series of principal submatrices of $k^2 \times k^2$. Considering that an SPD matrix is often defined by covariance in computer vision, we can follow this probabilistic clue to select principal submatrices.

Specifically, according to statistics, for a given image set consisting of images $X_1, \cdots, X_m$ of the size of $n \times n$, it can be viewed as m samples from a population, an $n \times n$ random matrix X. Then we estimate population covariance by sample covariance and model it into an $n^2 \times n^2$ SPD matrix S_0. After forward passing the deep SPD networks, we obtain a lower-dimensional discriminative SPD matrix S of $d^2 \times d^2$ via the network mapping denoted by $\phi(\cdot) : \mathcal{S}_{++}^{n^2} \rightarrow \mathcal{S}_{++}^{d^2}$. We hypothesize that there is an implicit mapping $\psi(\cdot) : \mathbb{R}^{n\times n} \rightarrow \mathbb{R}^{d\times d}$ to transform the random matrix X into another one $\psi(X)$ and S is the covariance of $\psi(X)$. We denote $\psi(X)$ as Y for simplicity and then the size of Y is $d \times d$. If we focus on the local region covariance of Y, then we can extract a series of principal submatrices from S. Besides, the number of submatrices extracted in this way is much smaller than the combinatorial number.

More specifically, we view a $d^2 \times d^2$ SPD matrix S as a covariance from a $d \times d$ random matrix Y, The covariance matrix associated with a $k \times k$ receptive field of the matrix Y corresponds to a $k^2 \times k^2$ principal submatrix of S. When moving a $k \times k$ sliding window by a step equal to s in Y (to obtain the position index), we can select $(\frac{d-k}{s} + 1)^2$ principal submatrices of S accordingly. Obviously, the submatrices we select are still SPD and the number of them is much smaller than $C_{d^2}^{k^2}$. Besides, it will encode the geometrical information conveyed by the local system in the category of SPD manifold. Figure 2 provides a conceptual illustration of the process of selecting submanifolds.

Multi-scale Mechanism In fact, in a deep SPD network, hidden SPD feature contains statistically compact information. Each element of the hidden feature reflects the region correlation in the original video. By extracting principal submatrices, we focus on the correlation among multiple local regions. In this way, we can capture the local semantic information in a statistical form. We anticipate that the classification can benefit from this compact local statistical information.

Furthermore, inspired by Res2Net (Gao et al. 2019a), which attempts to capture multi-scale features in one single residual block, we design a multi-channel mechanism to obtain local manifold information at diverse granularity levels. In detail, for an SPD feature, various BiMap-ReEig layers are applied to obtain a low dimensional SPD matrix, by:

$$\text{BiMap: } S_{k-1}^i = W_{k-1}^i S_{k-2} W_{k-1}^{iT}, \tag{3}$$

$$\text{ReEig: } S_k^i = U_{k-1}^i \max(\epsilon I, \Sigma_{k-1}^i) U_{k-1}^{iT}, \tag{4}$$

where all W_{k-1}^i are of the same size, and (4) is an eigenvalues function.

It is expected that in each channel, the primary learnt local geometry S_k^i varies. To capture it, submatrices with specific $k_i \times k_i$ dimensions are extracted from S_k^i. In this way, rich statistical local information at different granularity levels is extracted by our network. Besides, as an SPD matrix itself can be viewed as a trivial submanifold, we also incorporate a channel capturing the global information. In this way, not only holistic geometry, but also local geometry are jointly captured in the category of SPD manifold.

Submanifolds Fusion Strategies If we follow the Proposition 3 rigorously, it would result in a major computational burden, in view of the complex structure of the Riemannian manifold. Therefore, we fuse this multi-scale information in an approximate way instead. Since the SPD manifold $\mathcal{S}_{++}^d$ is diffeomorphic to the Euclidean space $\mathcal{S}^d$ of symmetric matrices by matrix logarithm, we can fuse the multi-scale information in a comparatively simple space, *i.e.*, $\mathcal{S}^d$, which is a well-known technique for tackling the problems of data analysis on SPD manifolds (Huang et al. 2015c).

To be more specific, we first map each principal submatrix into a real symmetric matrix by matrix logarithm, $M_k^{ij} = \log(P_k^{ij})$, where k, i and j represent k^{th} layer,i^{th} channel, and j^{th} submatrix respectively. Considering that the dimensionality of the Euclidean space formed by symmetric matrices $\mathcal{S}^d$ is $\frac{d(d+1)}{2}$, for each M_k^{ij}, we exploit its lower triangular part to further mitigate the computational burden. Since at this point the features lie in a Euclidean space, concatenation can be applied to fuse them. This process can be formally described as:

$$V_k^i = \text{concat}(\text{vec}(\text{tril}(M_k^{i1})), \cdots, \text{vec}(\text{tril}(M_k^{in^i}))) \tag{5}$$

where $\text{vec}(\cdot)$ means vectorization.

After we fuse the local information in each channel, we concatenate the feature vectors to aggregate the multi-scale information in different channels.

5 Experiments

We evaluate the proposed MSNet in three challenging visual classification tasks: video-based action recognition with the Cambridge-Gesture (CG) (Kim and Cipolla 2008) and the UCF-101 (Soomro, Zamir, and Shah 2012) datasets, and skeleton-based action recognition with the First-Person Hand Action (FPHA) (Garcia-Hernando et al. 2018) dataset, respectively. We simply employ a fully connected layer, softmax layer and cross-entropy as our output layer as (Huang and Van Gool 2017; Huang, Wu, and Van Gool 2018). For training our MSNet, we use an i5-9400 (2.90GHz) CPU with 8GB RAM.

5.1 Implementation Details

The SOTA Riemannian learning competitors include: *1). General methods for SPD learning:* Covariance Discriminative Learning (CDL) (Wang et al. 2012), SPD Manifold Learning (SPDML-AIM, SPDML-Stein) (Harandi, Salzmann, and Hartley 2018) and Log-Euclidean Metric Learning (LEML) (Huang et al. 2015c); *2). General methods for Grassmann learning:* Grassmann Discriminant Analysis (GDA) (Hamm and Lee 2008) and Projection Metric Learning (PML) (Huang et al. 2015b); *3). Hybrid Riemannian manifold learning methods:* Hybrid Euclidean-and-Riemannian Metric Learning (HERML) (Huang et al. 2015a) and Multiple Manifolds Metric Learning (MMML) (Wang et al. 2018); *4). Riemannian deep methods:* SPD Network (SPDNet) (Huang and Van Gool 2017), SymNet (Wang, Wu, and Kittler 2021), Grassmannian Network (GrNet) (Huang, Wu, and Van Gool 2018). All the comparative methods are carefully re-implemented by the source codes and fine-tuned according to the original papers.

To further evaluate the effectiveness of our algorithm, we also compare our MSNet with conventional SOTA hand pose estimation methods on the FPHA dataset. These approaches include Lie Group (Vemulapalli, Arrate, and Chellappa 2014), Hierarchical Recurrent Neural Network (HBRNN) (Du, Wang, and Wang 2015), Jointly Learning Heterogeneous Features (JOULE) (Hu et al. 2015), Convolutional Two-Stream Network (Two stream) (Feichtenhofer, Pinz, and Zisserman 2016), Novel View (Rahmani and Mian 2016), Transition Forests (TF) (Garcia-Hernando and Kim 2017), Temporal Convolutional Network (TCN) (Kim and Reiter 2017), LSTM (Garcia-Hernando et al. 2018) and Unified Hand and Object Model (Tekin, Bogo, and Pollefeys 2019). Besides, we also compare our approach against Euclidean network searching methods, DARTS(Liu, Simonyan, and Yang 2018) and FairDARTS (Chu et al. 2020), following the setting in (Sukthanker et al. 2021) by viewing SPD logarithm maps as Euclidean data.

We study five configurations, *i.e.*, MSNet-H, MSNet-PS, MSNet-AS, MSNet-S and MSNet-MS to further evaluate the utility of our proposed network. In detail, MSNet-MS with two BiMap-ReEig layers is $S_0 \rightarrow f_b^{(1)} \rightarrow f_r^{(2)} \rightarrow f_b^{(3)} \rightarrow f_r^{(4)} \rightarrow f_m^{(5)} \rightarrow f_f^{(6)} \rightarrow f_s^{(7)}$, where $f_b^{(k)}, f_r^{(k)}, f_m^{(k)}, f_f^{(k)}, f_s^{(k)}$ represent k^{th} layers of BiMap, ReEig, multi-scale submanifold, FC, and softmax-cross-entropy, respectively. Note that apart from the two BiMap-ReEig blocks in the backbone, in each branch of $f_m^{(k)}$, there is a BiMap-ReEig block as well, as illustrated in Figure 1. Besides, as the whole SPD matrix can be viewed as a trivial submanifold, we use MSNet-H to denote that we only extract holistic information. Though it is similar to SPDNet, only the lower triangular part is exploited for classification in our framework, alleviating the computational burden. To study the utility of proper submanifolds, we use MSNet-PS to represent that we extract all kinds of proper submanifolds according to our principles except the trivial ones. To see whether over-loaded submanifolds would bring about redundant information, we build MSNet-AS to extract all the sub-

Dataset	CG	FPHA	UCF-sub
BiMap Settings	100,80,50,25	63,56,46,36	100,80,49
Submanifolds	$2^2, 3^2, 4^2, 5^2$	$5^2, 6^2$	$2^2, 6^2, 7^2$
Learning Rates	$1e^{-5}$	$1e^{-4}$	$1e^{-5}$
Epochs	500	3500	500

Table 1: Configurations of MSNet on three datasets. Note that 100,80,50,25 means $100 \times 80, 80 \times 50, 50 \times 25$.

manifolds including the trivial ones. MSNet-S denotes that we only utilize the proper submanifolds in the corresponding MSNet-MS except for the trivial one.

In the experiments, we simply set "step" size equal to 1. The above models of our MSNet and SPDNet share the same learning mechanism as follows. The initial learning rate is $\lambda = 1e^{-2}$ and reduced by 0.8 every 50 epochs to a minimum of $1e^{-3}$. Besides, the batch size is set to 30, and the weights in BiMap layers are initialized as random semi-orthogonal matrices. For activation threshold in ReEig and dimension of transformation matrices in BiMap, we first search the optimal settings by our backbone SPDNet and then employ the same settings to our MSNet.

5.2 Datasets and Settings

To evaluate our method when facing limited data, experiments are carried out on the CG (Kim and Cipolla 2008) dataset. It consists of 900 video sequences covering nine kinds of hand gestures. For this dataset, following the criteria in (Chen et al. 2020), we randomly select 20 and 80 clips for training and testing per class, respectively. For evaluation, we resize each frame into 20×20 and obtain the grey scale feature. To further facilitate our experiment, we reduce each frame dimension to 100 by PCA. Then we compute the covariance matrix of size 100×100 to represent each video.

We employ the popular FPHA (Garcia-Hernando et al. 2018) dataset for skeleton-based action recognition. It includes 1,175 action videos of 45 different action categories. For a fair comparison, we follow the protocols in (Wang, Wu, and Kittler 2021). In detail, we use 600 action clips for training and 575 for testing and each frame is vectorized into a 63-dimensional feature vector with the provided 3D coordinates of 21 hand joints. Then we obtain a 63×63 covariance representation for each sequence.

To assess the utility of our method in the task of relatively large scale, UCF-101 (Soomro, Zamir, and Shah 2012) dataset is exploited, which is sourced from YouTube, containing 13k realistic user-uploaded video clips of 101 types of action. To facilitate our experiment, 50 kinds of action are randomly selected, each of which consists of 100 clips. We call this dataset UCF-sub in the following. As we did in the CG dataset, we exploit the grey scale feature and reduce the dimension of each frame to 100 by PCA. The seventy-thirty-ratio (STR) protocol is exploited to build the gallery and probes.

The configurations on three datasets are listed in Table 1.

Methods	Year	Colour	Depth	Pose	Acc.
Lie Group	2014	✗	✗	✓	82.69
HBRNN	2015	✗	✗	✓	77.40
JOULE	2015	✓	✓	✓	78.78
Two stream	2016	✓	✗	✗	75.30
Novel View	2016	✗	✓	✗	69.21
TF	2017	✗	✗	✓	80.69
TCN	2017	✗	✗	✓	78.57
LSTM	2018	✗	✗	✓	80.14
H+O	2019	✓	✗	✗	82.43
DARTS	2018	✗	✗	✓	74.26
FairDARTS	2020	✗	✗	✓	76.87
SPDML-AIM	2018	✗	✗	✓	76.52
HERML	2015	✗	✗	✓	76.17
MMML	2018	✗	✗	✓	75.05
SPDNet	2017	✗	✗	✓	85.57
GrNet	2018	✗	✗	✓	77.57
SymNet	2021	✗	✗	✓	82.96
MSNet-H		✗	✗	✓	85.74
MSNet-PS		✗	✗	✓	80.52
MSNet-AS		✗	✗	✓	82.26
MSNet-S		✗	✗	✓	86.61
MSNet-MS		✗	✗	✓	**87.13**

Table 2: Recognition Results (%) on the FPHA Dataset.

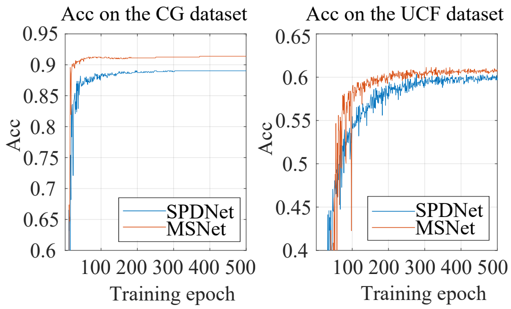

Figure 3: Accuracy curve of the proposed MSNet against SPDNet on the CG and UCF-sub datasets.

5.3 Analysis

As reported in Table 2 and Table 3, our proposed MSNet-MS outperforms all the other competitors on three datasets. Note that in a relatively large dataset, some shallow learning methods, such as MMML and HERML, are barely feasible in view of their time-consuming optimization. Figure 3 shows the convergence behaviour on the CG and UCF-sub datasets. There are several interesting observations worth discussing.

Firstly, although hybrid Riemannian methods, like HERML and MMML, can take advantage of complementary information encoded in different manifold and thus surpass some deep methods occasionally, our method consistently outperforms them. This verifies that information encoded in submanifolds is of great importance and could be beneficial for classification.

Secondly, on the CG and FPHA datasets, MSNet-H achieves almost the same performance as SPDNet, while on

Method	CG	UCF-sub
GDA	88.68	43.67
CDL	90.56	41.53
PML	84.32	50.60
LEML	71.15	44.67
SPDML-Stein	82.62	51.40
SPDML-AIM	88.61	51.13
HERML	88.94	NA
MMML	89.92	NA
GrNet	85.69	35.80
SPDNet	89.03	59.93
SymNet	89.81	56.73
MSNet-H	89.03	58.27
MSNet-PS	90.14	57.73
MSNet-AS	NA	58.33
MSNet-S	90.14	59.40
MSNet-MS	**91.25**	**60.87**

Table 3: Performance (%) on the CG and UCF-sub datasets.

the UCF-sub dataset, MSNet-H is inferior to SPDNet. Although as we discussed in Section 4.3, the lower triangular part of a symmetric matrix is mathematically equivalent to itself, the non-convexity of optimization on deep learning might cause some empirical deviation. However, extracting lower triangular, which is the way we exploit, could alleviate the computational burden and thus enhance the scalability of the output layer. For instance, if a deep network is used as an output classifier, it would be more efficient to halve the dimensionality of the input vector. What's more, our method, *i.e.*, MSNet-MS, surpasses the backbone network, SPDNet, in all three datasets. This suggests that the underlying statistical information in submanifolds could contribute to the visual perception, leading to a better classifier. Therefore, efforts should be paid to mine the information in submanifolds. However, over-saturated efforts in submanifolds might undermine discriminability. This idea is justified by the generally inferior performance of MSNet-AS, which selects all the submanifolds, to MSNet-MS.

Thirdly, although proper efforts should be made for submanifolds, they might vary for different tasks. On the CG dataset, the best performance is achieved when we use all kinds of submanifolds. This might be attributed to the particularity of the dataset. In detail, on this dataset, the background and the foreground are relatively monotonous, and the difference between them is apparent. Therefore, statistical information at diverse granular levels encoded in different submanifolds could contribute to the classification. However, on the FPHA and UCF-sub datasets, the large variance of appearance makes us cautious to select submanifolds to waive the statistically dispensable information.

More importantly, although local statistical information is of great value for visual classification, never can we neglect the importance of holistic information. Specifically, as we can see, MSNet-MS is superior to MSNet-S. The sole difference between these two configurations is that MSNet-MS uses the global information, while MSNet-S does not. The consistent phenomenon can be observed in the comparison between MSNet-AS and MSNet-PS. This indicates that only when we combine the optimal local information together with global information, could we make the best of statistical information.

Configuration	SPDNet	MSNet
{100,60,36}	56.73	57.33
{100,80,60,36}	56.27	58.33
{100,80,50,25}	48.47	52.67
{100,80,50,25,16}	40.07	45.60
{100,80,50,25,9}	28.73	36.07

Table 4: Performance (%) on the UCF-sub dataset under different backbone configurations.

It takes about 1.29s, 2.67s and 34.50s per epoch to train our MSNet-MS on the CG, FPHA and UCF-sub datasets, respectively, while training SPDNet takes 0.53s, 1.53s and 11.33s per epoch. Although the extra time caused by our multi-branch mechanism is inevitable, our method demonstrates the significance of submanifold for visual perception.

5.4 Ablation Study

To further evaluate the utility of our MSNet, different configurations, like depth and transformation matrices in SPD backbone, are implemented on the UCF-sub dataset, as shown in Table 4. Apart from the expected consistent performance gain brought about by our submanifold block, there is another interesting observation. The magnitude of improvement varies under different configurations. To be more specific, in some configurations, like {100,60,36}, the improvement sourced from our MSNet is relatively marginal, while in other cases, our approach offers more incremental gain, especially in the case of {100,80,50,25,9}, where the SPDNet is highly underfitting. This indicates that by providing complementary geometrically local information, submanifold is not only beneficial for Riemannian deep learning and could alleviate underfitting. It is therefore expected that the study of submanifold is worthwhile in the sense of promoting Riemannian deep learning forward.

6 Conclusion

In this paper, we successfully identify local mechanisms in Riemannian manifolds and propose a novel multi-scale submanifold block for SPD networks. Extensive experiments demonstrate the superiority of our approach. To the best of our knowledge, this work is the first attempt to mine the diverse local geometry in the Riemannian deep network paradigm. It opens a new direction that mines the information in a high-level semantic submanifold.

However, there are still some issues to be improved. For instance, our manual principle for selection is a sub-optimal expedient. In the future, we will explore other techniques for better selection. In addition, although local patterns are successfully defined, we rely on approximation in extracting local information. In the future, we will explore other more intrinsic ways to better deal with submanifolds.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U1836218, 62020106012, 61672265, 62106089), the 111 Project of Ministry of Education of China (B12018), and the Engineering and Physical Sciences Research Council (EPSRC) (EP/N007743/1, MURI/EPSRC/DSTL, EP/R018456/1).

References

Arsigny, V.; Fillard, P.; Pennec, X.; and Ayache, N. 2005. *Fast and Simple Computations on Tensors with Log-Euclidean Metrics.* Ph.d. diss., INRIA.

Belongie, S.; Malik, J.; and Puzicha, J. 2002. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4): 509–522.

Brooks, D.; Schwander, O.; Barbaresco, F.; Schneider, J.-Y.; and Cord, M. 2019. Riemannian Batch Normalization for SPD Neural Networks. In *Advances in Neural Information Processing Systems*, volume 32.

Chakraborty, R. 2020. ManifoldNorm: Extending Normalizations on Riemannian Manifolds. arXiv:2003.13869.

Chakraborty, R.; Bouza, J.; Manton, J.; and Vemuri, B. C. 2020. Manifoldnet: A Deep Neural Network for Manifold-valued Data with Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chakraborty, R.; Yang, C.-H.; Zhen, X.; Banerjee, M.; Archer, D.; Vaillancourt, D.; Singh, V.; and Vemuri, B. 2018. A Statistical Recurrent Model on the Manifold of Symmetric Positive Definite Matrices. *Advances in Neural Information Processing Systems*, 31.

Chen, K.-X.; Ren, J.-Y.; Wu, X.-J.; and Kittler, J. 2020. Covariance Descriptors on a Gaussian Manifold and Their Application to Image Set Classification. *Pattern Recognition*, 107: 107463.

Chen, Z.; Xu, T.; Wu, X.-J.; Wang, R.; and Kittler, J. 2021. Hybrid Riemannian Graph-Embedding Metric Learning for Image Set Classification. *IEEE Transactions on Big Data*.

Chu, X.; Zhou, T.; Zhang, B.; and Li, J. 2020. Fair Darts: Eliminating Unfair Advantages in Differentiable Architecture Search. In *Proceedings of the European Conference on Computer Vision*, 465–480.

Du, Y.; Wang, W.; and Wang, L. 2015. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1110–1118.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional Two-stream Network Fusion for Video Action Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1933–1941.

Gao, S.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; and Torr, P. H. 2019a. Res2net: A New Multi-scale Backbone Architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gao, Z.; Wu, Y.; Harandi, M.; and Jia, Y. 2019b. A Robust Distance Measure for Similarity-based Classification on the SPD Manifold. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9): 3230–3244.

Garcia-Hernando, G.; and Kim, T.-K. 2017. Transition Forests: Learning Discriminative Temporal Transitions for Action Recognition and Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 432–440.

Garcia-Hernando, G.; Yuan, S.; Baek, S.; and Kim, T.-K. 2018. First-person Hand Action Benchmark with RGB-D Videos and 3d Hand Pose Annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 409–419.

Guilleminot, J.; and Soize, C. 2012. Generalized Stochastic Approach for Constitutive Equation in Linear Elasticity: a Random Matrix Model. *International Journal for Numerical Methods in Engineering*, 90(5): 613–635.

Hamm, J.; and Lee, D. D. 2008. Grassmann Discriminant Analysis: a Unifying View on Subspace-based Learning. In *International conference on machine learning*, 376–383.

Harandi, M.; Salzmann, M.; and Hartley, R. 2018. Dimensionality Reduction on SPD Manifolds: The Emergence of Geometry-Aware Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1): 48–62.

Harandi, M. T.; Sanderson, C.; Hartley, R.; and Lovell, B. C. 2012. Sparse Coding and Dictionary Learning for Symmetric Positive Definite Matrices: A Kernel Approach. In *European Conference on Computer Vision*, 216–229. Springer.

Hu, J.-F.; Zheng, W.-S.; Lai, J.; and Zhang, J. 2015. Jointly Learning Heterogeneous Features for RGB-D Activity Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5344–5352.

Hua, X.; Cheng, Y.; Wang, H.; Qin, Y.; Li, Y.; and Zhang, W. 2017. Matrix CFAR Detectors Based on Symmetrized Kullback–Leibler and Total Kullback–Leibler Divergences. *Digital Signal Processing*, 69: 106–116.

Huang, Z.; and Van Gool, L. 2017. A Riemannian Network for SPD Matrix Learning. In *Thirty-first AAAI conference on artificial intelligence*.

Huang, Z.; Wang, R.; Shan, S.; and Chen, X. 2015a. Face Recognition on Large-scale Video in the Wild with Hybrid Euclidean-and-Riemannian Metric Learning. *Pattern Recognition*, 48(10): 3113–3124.

Huang, Z.; Wang, R.; Shan, S.; and Chen, X. 2015b. Projection Metric Learning on Grassmann Manifold with Application to Video Based Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 140–149.

Huang, Z.; Wang, R.; Shan, S.; Li, X.; and Chen, X. 2015c. Log-Euclidean Metric Learning on Symmetric Positive Definite Manifold with Application to Image Set Classification. In *International Conference on Machine Learning*, 720–729. PMLR.

Huang, Z.; Wu, J.; and Van Gool, L. 2018. Building Deep Networks on Grassmann Manifolds. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kim, T.-K.; and Cipolla, R. 2008. Canonical Correlation Analysis of Video Volume Tensors for Action Categorization and Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8): 1415–1428.

Kim, T. S.; and Reiter, A. 2017. Interpretable 3D Human Action Analysis with Temporal Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1623–1631.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25.

Kulis, B.; Sustik, M.; and Dhillon, I. 2006. Learning Low-rank Kernel Matrices. In *International Conference on Machine Learning*, 505–512.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lin, Z. 2019. Riemannian Geometry of Symmetric Positive Definite Matrices via Cholesky Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 40(4): 1353–1370.

Liu, H.; Simonyan, K.; and Yang, Y. 2018. Darts: Differentiable Architecture Search. arXiv:1806.09055.

Lowe, D. G. 2004. Distinctive Image Features from Scale-invariant Keypoints. *International Journal of Computer Vision*, 60(2): 91–110.

Nguyen, X. S. 2021. GeomNet: A Neural Network Based on Riemannian Geometries of SPD Matrix Space and Cholesky Space for 3D Skeleton-Based Interaction Recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 13379–13389.

Pennec, X.; Fillard, P.; and Ayache, N. 2006. A Riemannian Framework for Tensor Computing. *International Journal of Computer Vision*, 66(1): 41–66.

Rahmani, H.; and Mian, A. 2016. 3D Action Recognition from Novel Viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1506–1515.

Roman, S.; Axler, S.; and Gehring, F. 2005. *Advanced Linear Algebra*, volume 3. Springer.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: a Dataset of 101 Human Actions Classes from Videos in the Wild. arXiv:1212.0402.

Sukthanker, R.; Huang, Z.; Kumar, S.; Endsjo Goron, E.; Wu, Y.; and Van Gool, L. 2021. Neural Architecture Search of SPD Manifold Networks. In *International Joint Conferences on Artificial Intelligence*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going Deeper with Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Tekin, B.; Bogo, F.; and Pollefeys, M. 2019. H+O: Unified Egocentric Recognition of 3D Hand-object Poses and Interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4511–4520.

Tu, L. W. 2011. *An Introduction to Manifolds*. Springer.

Vemulapalli, R.; Arrate, F.; and Chellappa, R. 2014. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 588–595.

Wang, R.; Guo, H.; Davis, L. S.; and Dai, Q. 2012. Covariance Discriminative Learning: A Natural and Efficient Approach to Image Set Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2496–2503. IEEE.

Wang, R.; Wu, X.-J.; Chen, K.-X.; and Kittler, J. 2018. Multiple Manifolds Metric Learning with Application to Image Set Classification. In *International Conference on Pattern Recognition*, 627–632. IEEE.

Wang, R.; Wu, X.-J.; Chen, Z.; Hu, C.; and Kittler, J. 2022a. SPD Manifold Deep Metric Learning for Image Set Classification. *IEEE Transactions on Neural Networks and Learning Systems*.

Wang, R.; Wu, X.-J.; Chen, Z.; Xu, T.; and Kittler, J. 2022b. DreamNet: A Deep Riemannian Manifold Network for SPD Matrix Learning. In *Proceedings of the Asian Conference on Computer Vision*, 3241–3257.

Wang, R.; Wu, X.-J.; and Kittler, J. 2021. Symnet: A Simple Symmetric Positive Definite Manifold Deep Learning Method for Image Set Classification. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5): 2208–2222.

Zhang, T.; Zheng, W.; Cui, Z.; Zong, Y.; Li, C.; Zhou, X.; and Yang, J. 2020. Deep Manifold-to-Manifold Transforming Network for Skeleton-Based Action Recognition. *IEEE Transactions on Multimedia*, 22(11): 2926–2937.

Zhen, X.; Chakraborty, R.; Vogt, N.; Bendlin, B. B.; and Singh, V. 2019. Dilated Convolutional Neural Networks for Sequential Manifold-valued Data. In *Proceedings of the IEEE International Conference on Computer Vision*, 10621–10631.