Artificial Intelligence and Human-Computer Interaction
Y. Ye and P. Siarry (Eds.)

doi:10.3233/FAIA240133

Learning Submanifold-Specific Normalization and Attention with SPD Matrices for Visual Classification

Enxu LV [a,b], Hengliang TAN [a,b,1], Jiao DU [a], Shuo YANG [a], Guofeng YAN [a] and Yijian ZHAO [a,b]
[a] *School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China*
[b] *Metaverse Research Institute, Guangzhou University, Guangzhou, China*
ORCiD ID: Hengliang Tan https://orcid.org/0000-0003-2167-156X

Abstract. Deep manifold learning has achieved significant success in handling visual tasks by using Symmetric Positive Definite (SPD) matrices, particularly within multi-scale submanifold networks (MSNet). This network is capable of extracting a series of main diagonal submatrices from SPD matrices. However, these submanifolds do not take into account the distribution of the submanifolds themselves. To address this limitation and introduce batch normalization tailored to submanifolds, we devise a submanifold-specific normalization approach that incorporates submanifold distribution information. Additionally, for submanifolds mapped into Euclidean space, considering the weight relationships between different submanifolds, we propose an attention mechanism tailored for log mapped submanifolds, termed submanifold attention. Submanifold attention is decomposed into multiple 1D feature encodings. This approach enables the capture of dependencies between different submanifolds, thus promoting a more comprehensive understanding of the data structure. To demonstrate the effectiveness of this method, we conducted experiments on various visual databases. Our results indicate that this approach outperforms the MSNet.

Keywords. Manifold Learning, attention mechanism, symmetric positive definite matrices, batch normalization

1. Introduction

In the studies of visual classification, the natural second-order statistic of covariance matrix has been proven to be successful in describing the visual feature. The non-singular covariance matrices known as the Symmetric Positive Definite (SPD) matrices which can form the SPD Riemannian manifold. Classification tasks with SPD manifold for video data have received increasing attention in recent years, such as video-based facial emotion recognition [1-4],dynamic scene classification [5-7],action recognition [8-12], and video-based face recognition [13-16].

Huang et al. introduced a deep network structure for SPD manifold [1], which reduces the dimensionality of the input data through bilinear mapping and maintains the

[1] Corresponding Author: Hengliang Tan, School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China. E-mail: tanhengliang@gzhu.edu.cn.

manifold's positive definiteness through a rectification module. Influenced by the Euclidean Batch Normalization [17] network, Brooks et al. utilized parallel transmission to design the Riemannian normalization network [18] for SPDNet. In contrast to SPDNet, Zhang et al. proposed a 2D convolution network for SPD matrices [19], requiring the convolution kernels to be SPD matrices. Inspired from U-Net [20], Wang et al. redesigned the covariance matrix to create the U-SPDNet [12] network, allowing the feature extraction component to learn more informative low-dimensional mappings. DreamNet [11] addressed the problem of poor performance in deepening the SPDNet by using a stacked U-SPDNet approach and reconstructing SPD matrices.

Chen [21] et al. proposed a method that utilizes SPD matrices by extracting sub-SPD matrices, thereby preserving the regularity and symmetry of SPD matrices while fully exploiting their intrinsic information. Features from SPD matrices of different scale sizes are extracted and classified using the softmax loss function. The multi-scale submanifold network involves extracting small submanifolds of different scales from the SPD manifold and then concatenating the upper triangular parts of these submanifolds into column vectors for classification. We found that the submanifolds extracted from the SPD matrix should be normalized, and different attention should be given to different submanifolds.

Inspired by the concept of Riemannian Batch Normalization (BN) for SPD neural networks [18], we proposed a novel form of BN tailored specifically for SPD submanifolds, termed “group batch normalization”. Unlike conventional SPD BN, our designed group BN is applied individually to each submanifold.

In the realm of deep learning neural networks, attention mechanisms have played a pivotal role in enhancing the capacity to model complex relationships within data. Coordinate Attention (CA) [22] is an attention mechanism that embeds positional information into channel attention. It takes into account the spatial coordinates and positions of elements in a data structure, granting varying importance to different positions. CA has proven to be effectiveness in computer vision and spatial reasoning tasks. We draw inspiration from the CA [22] and devise an attention mechanism to cater to different submanifolds.

The main contributions of this paper can be summarized as follows:

- We devised a group normalization approach tailored for submanifold, where each distinct submanifold undergoes Riemannian normalization.
- We devised an attention mechanism inspired by CA, assigning different weights to different submanifolds by learning.
- We demonstrated the effectiveness of our method by extensive experimentation with different datasets.

The rest of this paper is organized as follows: we provide an overview of essential basic knowledge about Riemannian manifolds in Section 2. The proposed method is presented in Section 3. The validations of the proposed method are presented in Section 4. Section 5 concludes this paper.

2. Preliminaries

2.1. SPD manifold

The SPD manifold is a concept of significant relevance in differential geometry and manifold theory. For all non-zero vectors $v \in R^d$, a real-valued matrix M is termed an SPD matrix if and only if $v^T Mv > 0$. The SPD manifold comprises a collection of $d \times d$ matrices that satisfy this property, forming a commutative Lie group structure denoted as S^{d++}. The inherent properties of this manifold contribute to its widespread applications in fields such as differential geometry and data analysis.

2.2. Covariance matrix

The covariance matrix is used to describe the correlation information between features of classes within a dataset, given a set of images $X_i = \left[x_1, x_2, ..., x_{n_i}\right]$,The corresponding covariance is represented as

$$M_i = \frac{1}{n_i - 1}\sum_{k=1}^{n_i}(x_k - u_i)(x_k - u_i)^T \tag{1}$$

where u_i is the mean of X_i and the formula is given by: $u_i = \sum_{k=1}^{n_i} x_k$, where M_i is symmetric and potentially non-singular. By introducing elements on the diagonal, $M_i = M_i + \alpha I$, where α is a regularization coefficient, and I is an identity matrix. M_i can be rendered non-singular, making it represents an element of the SPD manifold.

2.3 Parallel transport

Theorem 1 [23]: Let $A, B, P \in M$ and let $S = \log_B(P) \in T_B^M$, then,

$$Exp_A(\Gamma_{B \to A}(S)) = EPE^T \tag{2}$$

where $E = (AB^{-1})^{1/2}$

Theorem 1 defines the parallel transport of points on a manifold. For a point P on a point manifold, it involves mapping the tangent space at point P to the tangent space at point B , then parallelly transporting it to the tangent space at point A , and finally mapping it back to the manifold space using the exponential function.

3. The proposed method

3.1. Riemannian local mechanism network

The Riemannian Local Mechanism Network (RLMN) is a classification network that operates as follows: It begins by employing a sequence of BiMap-ReEig layers for dimensionality reduction, which are subsequently connected in a series. Then, parallel BiMap-ReEig layers generate multiple SPD matrices of the same size. Each SPD matrix undergoes processing through a Subsec layer, yielding corresponding submanifold groups. The primary objective of this process is to extract submanifold groups of different scales. Following this, the LogEig layer maps these submanifold groups to the tangent space. Subsequently, the TrilCon layer extracts the upper triangular matrix elements from each submanifold group, assembling them into column vectors. These obtained column vectors are concatenated using the Concat layer, and a softmax classifier is employed for classification.

The BiMap layer transforms the input SPD matrices by using a bilinear mapping, enhancing their distinctiveness and compactness.

$$X^{(L)} = WX^{(L-1)}W^{T} \tag{3}$$

where $X^{(L-1)}$ represents the input SPD matrix, $X^{(L)}$ is the output SPD matrix, and W denotes the transformation matrix W is an orthogonal matrix and possesses full column rank. The ReEig layer is similar to the activation function in CNN networks, the obtained eigenvalues are subject to an activation function following their decomposition. Afterward, they are reassembled as follows:

$$X^{(L)} = U^{(L-1)} max(\grave{o}I, \Sigma^{(L-1)}) U^{(L-1)T} \tag{4}$$

$$X^{(L)} = U^{(L-1)} \Sigma^{(L-1)} U^{(L-1)T} \tag{5}$$

where $U^{(L-1)}$ represents the matrix obtained from eigenvalue decomposition of the input SPD matrix $X^{(L)}$, $\Sigma^{(L-1)}$ denotes the eigenvalue matrix obtained from the SPD matrix decomposition, and ò is the activation threshold.

The LogEig layer shares a similar structure to the ReEig layer. It is used to map the SPD matrices to the tangent space, enabling the use of classifiers from Euclidean space for classification purposes. This is achieved by applying a logarithm operation to the eigenvalues through eigenvalue decomposition.

$$X^{(L)} = U^{(L-1)} log(\Sigma^{L-1}) U^{(L-1)T} \tag{6}$$

$$X^{(L)} = U^{(L-1)} \Sigma^{(L-1)} U^{(L-1)T} \tag{7}$$

The Subsec layer extracts principal submatrices along the diagonal direction, resulting in multiple SPD matrices corresponding to distinct submanifolds.

In the TrilCon layer, the upper triangular of the SPD matrices from each submanifold are extracted to form column vectors. Valuable information can be effectively captured by extracting all elements of upper triangular of SPD matrix.

The Concat layer combines column vectors from different scales into a single column vector, facilitating linear dimension reduction for subsequent Fully Connected (FC) layers.

The FC layer performs linear dimension reduction by transforming the input matrix $X^{(L-1)}$ into the output matrix $X^{(L)}$ by using the transformation matrix W.

$$X^{(L)} = WX^{(L-1)} \tag{8}$$

The Softmax layer is utilized at the end, and the chosen objective function is the cross-entropy loss, serving as the final loss function.

$$f(z_k) = \frac{e^{z_k}}{\sum_j e^{z_j}} \tag{9}$$

$$l(y,z) = -\sum_{k=0}^{c} y_k \log(f(z_k)) \tag{10}$$

where $f(z_k)$ represents the output value of the softmax function, and y_k denotes the actual class label.

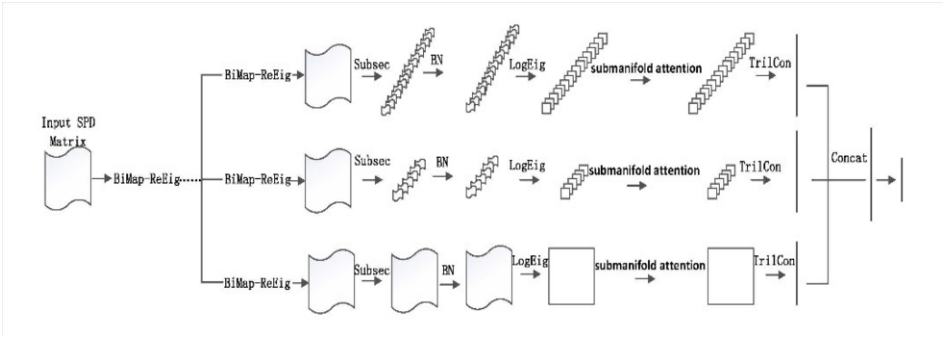

Figure 1. The architecture of the proposed method.

The architecture of the RLMN (also known as the Multi-scale Submanifold Network (MSNet) [19]) can be summarized as follows:

$$X_0 \to f_b \to f_r \to f_b \to f_r \to multiple f_b f_r \to Subsec \to LogEig \to TrilCon \to Concat \to f_c \to f_{softmax}$$

where X_0 represents the input SPD matrix, f_b denotes the BiMap layer, f_r denotes the ReEig layer, multiple f_b refers to the parallel application of BiMap layers on the same SPD matrix, Subsec indicates the extraction of multiple submanifolds from a single

SPD matrix, LogEig represents the mapping of submanifolds to Euclidean space, TrilCon involves extracting upper triangular elements from SPD matrices and forming column vectors, Concat signifies the concatenation of column vectors extracted from the previous layer into a single column vector, f_c is the FC layer, $f_{softmax}$ applies the softmax function.

In this study, we modify the MSNet by adding the submanifold batch normalization behind the Subsec layer, and the submanifold-specific attention is placed before the TrilCon layer. Figure 1 shows the architecture of the proposed network.

3.2. Submanifold-specific batch normalization

For the submanifold groups extracted from the SPD matrices in the aforementioned MSNet network, optimal results are challenging to be obtained due to the lack of knowledge about the data distribution within each submanifold group. Therefore, it is beneficial to apply separate batch normalization to each submanifold group. This allows the network to learn the data distribution information specific to each submanifold group.

Algorithm 1 Algorithm for computing the Karcher mean of multiple SPD matrix [24]

Require: Set of k SPD matrix $\{\mathrm{M}_c\}_{c=1}^{K}$

Ensure: Karcher mean μ

The arithmetic mean of a set of SPD matrices as inital estimate μ_1

$$\mu_1 = \frac{1}{K}\sum_{c=1}^{K} M_c$$

Compute logarithmic map

$$v_c = \log_{\mu_1}(M_c), \forall c = 1, \cdots, k$$

Compute average tangent vector

$$\hat{v} = \frac{1}{K}\sum_{c=1}^{K} v_c$$

Compute exponential map

$$\mu = \exp_{\mu_1}(\hat{v})$$

Algorithm 2 Submanifold group normalization on $\left\{M_{jk}\right\}_{j=1,\cdots,m}^{k=1,\cdots,n_m}$,training and testing phase

TRAINING PHASE

Require: A group of submanifold $\left\{M_{jk}\right\}_{j=1,\cdots,m}^{k=1,\cdots,n_m}$, mean $\mathrm{G}_1, \mathrm{G}_2, \cdots, \mathrm{G}_m$, bias $G_1, G_2, \cdots, G_m$

Ensure: Normalized submanifold $\widetilde{\mathrm{M}}_{jk}$

$$\mathrm{G}_j = Karcher(M_{j1}, M_{j2}, \cdots, M_{jn_j}) \quad j = 1, 2, \cdots, m$$

where m is the number of submanifold group

$$\mathrm{G}_j = Bar_{\mu}(\mathrm{G}_{sj}, \mathrm{G}_j),$$

G_{sj} represents the center of the j-th submanifold group

for $j = 1 \rightarrow m$ **do**

for $k = 1 \rightarrow n_m$ **do**

$$\overline{M_{jk}} = \Gamma_{\mathrm{G}_j \rightarrow I_d}(M_{jk})$$

$$\widetilde{M}_{jk} = \Gamma_{I_d \rightarrow G_j}(\overline{M_{jk}})$$

End for

End for

return normalized $\tilde{M}_{jk}$

TESTING PHASE

Require: A group of submanifold $\{M_{jk}\}_{j=1,\cdots,m}^{k=1,\cdots,n_m}$, mean $\mathbb{G}_1,\mathbb{G}_2,\cdots,\mathbb{G}_m$, bias $G_1,G_2,\cdots,G_m$

for $j=1\rightarrow m$ **do**

  for $k=1\rightarrow n_m$ **do**

$$\overline{M_{jk}}=\Gamma_{\mathbb{G}_j\rightarrow I_d}(M_{jk})$$

$$\tilde{M}_{jk}=\Gamma_{I_d\rightarrow G_j}(\overline{M_{jk}})$$

  End for

End for

return normalized $\tilde{M}_{jk}$

First, compute the Riemannian barycenter for a batch of data using the Karcher flow, the specific algorithm is illustrated in Algorithm 1. After obtaining the centroid for this batch, in order to achieve a more accurate centroid, the center of this batch's centroid and the previous centroid's position are used as the centroid points for this iteration. The specific formula for calculating the centroids of two SPD matrices is as follows:

$$Bar_{(w,1-w)}(P_1,P_2)=P_2^{1/2}(P_2^{-1/2}P_1P_2^{-1/2})^wP_2^{1/2} \tag{11}$$

Assuming there are j submanifold groups, each containing k submanifolds, the symbol corresponding to each submanifold can be denoted as $\{M_{jk}\}_{j=1,\cdots,m}^{k=1,\cdots,n_m}$, m represents the number of submanifold groups, and n_m represents the number of submanifolds contained within the m-th submanifold group. The mean for each submanifold group through Algorithm 1 can be denoteds: $\mathbb{G}_1,\mathbb{G}_2,\cdots,\mathbb{G}_m$.

The submanifold M_{jk} of each submanifold group is first projected onto the tangent space of the corresponding group's center point $\mathbb{G}_j$. Then, they are parallelly transported to the tangent space of the identity matrix, and subsequently mapped back to the submanifold to obtain the new submanifold $\tilde{M}_{jk}$. The formula for this process is as follows:

$$\overline{M_{jk}}=\Gamma_{\mathbb{G}_j\rightarrow I_d}(M_{jk})=\mathbb{G}_j^{-1/2}M_{jk}\mathbb{G}_j^{-1/2} \tag{12}$$

The process described above involves moving the submanifolds of each submanifold group to their respective center points by using parallel transport. Subsequently, they are transported to the bias by using parallel transport. The formula for this process is as follows:

$$\tilde{M}_{jk}=\Gamma_{I_d\rightarrow G_j}(\overline{M_{jk}})=G_j^{1/2}\overline{M_{jk}}G_j^{1/2} \tag{13}$$

The above process can be charactered by using Algorithm 2.

3.3. Submanifold-specific group attention menachim

The global pooling operation in SENet indeed captures global information. However, when it is applied individually to each submanifold group, it can only capture relationships within each specific group and may not capture interactions between different submanifold groups. Therefore, it is essential to capture information between different submanifold groups. To achieve this, we perform separate global average pooling (Avgpool) operations on submanifold groups of different scales.

The Avgpool formula for obtaining the average value of each submanifold can be described as:

$$Z_{w,h}^{j,k}(j,k)=\frac{1}{N_{jk}}\sum_{w,h}M(j,k,w,h) \tag{14}$$

where $M(j,k,w,h)$ represents the value of the k-th submanifold in the j-th submanifold group, N_{jk} denotes the total number of submanifold elements in the j-th submanifold group, and w and h correspond to the dimensions (length and width) of the submanifold.

Through the aforementioned steps, we can obtain attention information for different submanifold groups. To make full use of the resulting representation information, we need to perform a second transformation, referred to as "submanifold group attention generation." Our design objectives for this transformation are threefold: firstly, it should be computationally efficient; secondly, it should capture information between different submanifold groups; and finally, it should capture relationships between submanifolds within the same submanifold group.

The feature maps generated by Equation 14 are first subjected to an aggregation operation. Subsequently, a 1×1 shared convolution transformation denoted as F_1 is applied.

$$f=\sigma(F_1([z^{(1,1)},z^{(1,2)},\cdots,z^{(1,n_1)},\cdots,z^{(m,1)},\cdots,z^{(m,n_m)}])) \tag{15}$$

where m represents the index of submanifold groups, and n_m represents the number of submanifolds in the m-th submanifold group, the feature aggregation operation is represented by using the symbol "[,]". σ is the activation function. $f\in R^q, q=\sum_{k=1}^{m}n_k\left(\sum_{k=1}^{m}n_k\right)\Big/r$ is the lower dimensional feature map that reduced by the transformation F_1, r is the reduced dimension.

We partition the tensor f into m tensors by:

$$f^{n1},f^{n2},\cdots,f^{nm}=split(f) \tag{16}$$

The dimensions of $f^{n1}, f^{n2}, \cdots, f^{nm}$ are $\mathrm{r}_{n1}, \mathrm{r}_{n2}, \cdots, \mathrm{r}_{nm}$ respectively, it can be seen that the dimensions are satisfied $\sum_{k=1}^{m} r_{nk} = r$. $f^{n1} \in R^{r_{n1} \times 1}, f^{n1} \in R^{r_{n2} \times 1}, \cdots, f^{nm} \in R^{r_{nm} \times 1}$, $F_{n1}, F_{n2}, \cdots, F_{nm}$ are used to restore the tensor to the dimensions of the corresponding submanifolds.

$$g^{n1} = \delta(F_{n1}(f^{n1})) \tag{17}$$

$$g^{n2} = \delta(F_{n2}(f^{n2})) \tag{18}$$

$$g^{nm} = \delta(F_{nm}(f^{nm})) \tag{19}$$

where δ represents the sigmod function. To reduce to the appropriate dimension, it is necessary to choose the suitable value of r . The output tensor g_{nj} represents the weight of the j-th submanifold group, and it is multiplied with the corresponding submanifold group. Finally, the weighted submanifold can be expressed as:

$$y_{nj} = x_{nj} \cdot g^{nj} \quad j = 1, \cdots, m \tag{20}$$

where x_{nj} is the input of the j-th submanifold group.

Ultimately, this procedure enables the acquisition of weighted submanifolds that encapsulate enhanced expressive power for the extracted submanifold features, the architecture of the submanifold-specific attention mechanism is shown in Figure 2. And we call the proposed multi-scale submanifold-specific batch normalization and attention method as MSNet-bn-attention.

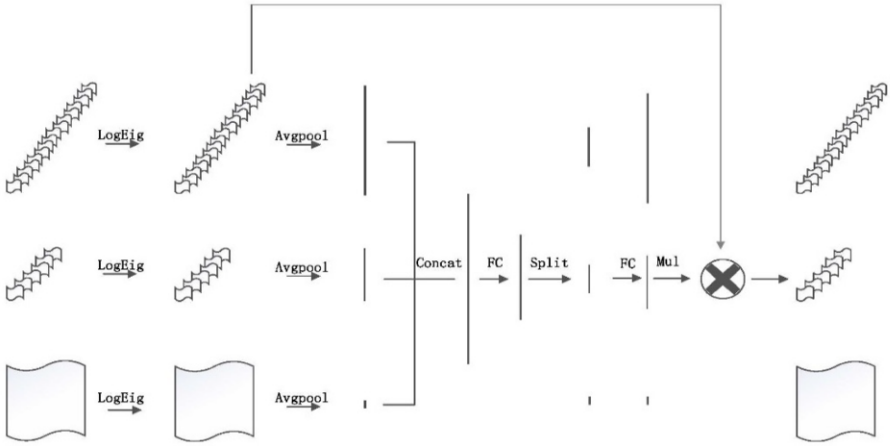

Figure 2. The submanifold-specific attention mechanism.

4. Experiments

In this section, we evaluated the proposed method on visual recognition tasks. Three datasets are utilized to our experiments, they are the CG dataset [25], the FPHA dataset [26], and the UAV dataset [27]. For the final classification, we employ fully connected layers and the softmax as the loss function. The experiments were conducted on an i7 CPU with 16GB of memory with Python. Extensive comparative methods are used for comparisons.

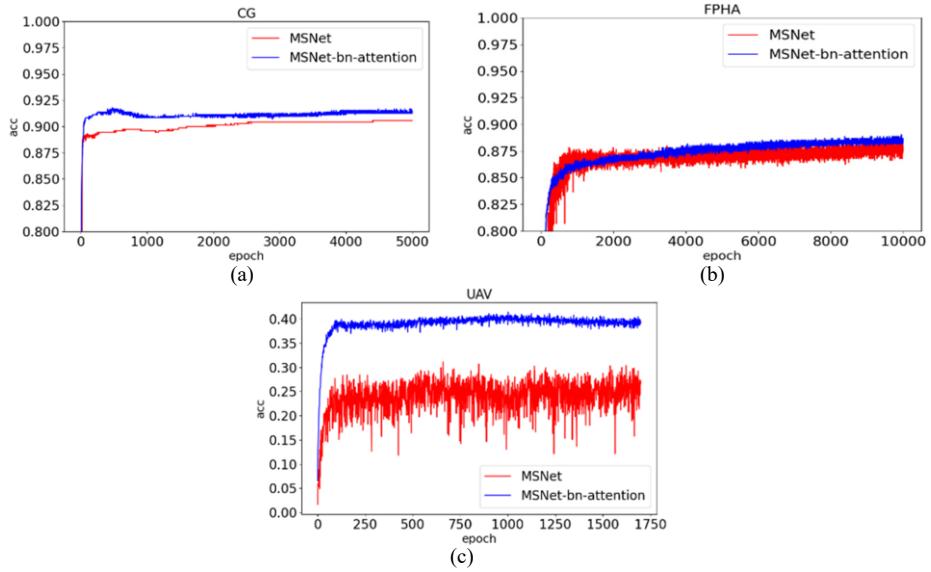

Figure 3. Performance comparison with different epochs on different datasets.

4.1. CG dataset

Cambridge-Gesture (CG) Dataset [25] comprises 900 image sequences, encompassing 9 gesture categories. These categories are defined by 3 fundamental gesture shapes and 3 fundamental actions. Thus, the objective of this dataset is to simultaneously classify different shapes and actions. To utilize the CG database, each frame's image was dimensionally reduced to 100 by PCA, yielding a 100×100 covariance matrix.

In this task, the sizes of BiMap weights are 100×80 and 80×50. To obtain submanifolds at different scales, the sizes of Bimap weights are 50×25, the multi-scale submanifolds result in submanifolds with length and width of 2, 3, 4, and 5, respectively, and the multi-scale submanifold extraction yields four groups of submanifolds, each with a respective count of 16, 9, 4, and 1, the dimension reduction ratios for the submanifolds in each submanifold group are 16/8, 9/6, 4/2, and 1/2 respectively.

We have rewritten the MSNet in Python and conducted comparative experiments with the proposed network MSNet-bn-attention. As depicted in Figure 3(a), comparing to the original MSNet, the proposed method always outperforms the MSNet in most of the 5000 epochs. As shown in the second column of Table 1, the MSNet with different scales [19] are also inferior to our method. Comparing to other methods, the proposed method achieves the best recognition rate.

4.2. FPHA dataset

First-Person Hand Action (FPHA) benchmark dataset [26] is a collection of RGB-D video sequences comprised of more than 100K frames of 45 daily hand action categories. FPHA involves 26 different objects in several hand configurations. To utilize the FPHA database, the image of each frame was dimensionally reduced to 63 by PCA, yielding a 63x63 covariance matrix.

Table 1. Recognition rates (%) comparison on different datasets.

Method	CG	FPHA	UAV
GDA [28]	88.68	N/A	28.13
CDL [15]	90.56	N/A	31.11
PML [14]	84.32	N/A	10.66
LEML [2]	71.15	N/A	N/A
SPDML-Stein [13]	82.62	N/A	N/A
SPDML-AIM [13]	88.61	76.20	N/A
HERML [29]	88.94	76.17	N/A
MMML [16]	89.92	75.05	N/A
GrNet [3]	85.69	77.57	35.23
SPDNet [1]	89.03	85.57	N/A
SymNet [30]	89.81	82.96	N/A
MSNet-H [21]	89.30	85.74	N/A
MSNet-PS [21]	90.14	80.52	N/A
MSNet-AS [21]	N/A	82.26	N/A
MSNet-S [21]	90.14	86.61	N/A
MSNet-MS [21]	91.25	87.13	N/A
Lie Group [31]	N/A	82.69	N/A
HBRNN [32]	N/A	77.4	N/A
JOULE [33]	N/A	78.78	N/A
Two stream [34]	N/A	75.30	N/A
Novel View [35]	N/A	69.21	N/A
TF [36]	N/A	80.69	N/A
TCN [37]	N/A	78.57	N/A
LSTM [26]	N/A	80.14	N/A
H+O [38]	N/A	82.43	N/A
DARTS [39]	N/A	74.26	36.13
FairDARTS [40]	N/A	76.87	40.01
MSNet (Python)	90.5	88.6	31.11
MRMML [41]	N/A	N/A	N/A
SPDML [13]	N/A	N/A	22.69
GEMKML [4]	N/A	N/A	N/A
ManifoldNet [42]	N/A	N/A	N/A
DeepO2P [43]	N/A	N/A	N/A
MSNet-bn-attention	**91.8**	**89.04**	**41.44**

In this task, the sizes of Bimap weights are 63×56 and 56×46. To obtain submanifolds at different scales, the sizes of Bimap weights are 46×36, the multi-scale submanifolds result in submanifolds with length and width of 5 and 6, respectively. The multi-scale submanifold extraction yields four groups of submanifolds, each with a respective count of 4 and 1. The dimension reduction ratios for the submanifolds in each submanifold group are 4/2, and 2/2 respectively.

We compared our method with the MSNet in Python as depicted in Figure 3(b). 10,000 epochs are evaluated in this experiment, the performance of MSNet is slightly higher than our MSNet-bn-attention before 2000 epochs, however our method outperforms MSNet in the epochs larger than 2000. This result demonstrates the effectiveness of the submanifold-specific batch normalization and attention mechanism.

Moreover, as shown in the third column of Table 1, our approach outperforms other methods.

4.3. UAV dataset

The UAV-Human (UAV) dataset [27] is designed for the understanding and analysis of human behavior in unmanned aerial vehicle imagery. It comprises a total of 67,428 multimodal video sequences and 119 targets for action recognition. Among these sequences, 22,476 frames are dedicated to pose estimation, 41,290 frames and 1,144 identities are used for person re-identification, and an additional 22,263 frames are allocated for attribute recognition.

In this scenario, each action video is described by an SPD matrix of size 51×51. Finally, the seventy-thirty-ratio (STR) protocol is applied 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, 43×36). To obtain submanifolds at different scales, the sizes of Bimap weights are 36×25 and 50x25. The multi-scale submanifolds result in submanifolds with length and width of 2, 3, 4, and 5, respectively. The multi-scale submanifold extraction yields four groups of submanifolds, each with a respective count of 16, 9, 4, and 1. The dimension reduction ratios for the submanifolds in each submanifold group are 16/8, 9/6, 4/2, and 1/2 respectively.

The rewritten algorithm of MSNet in Python is compared to the proposed method as depicted in Figure 3(c). From the result, we can see that our method achieves significant improvement comparing to the original MSNet. The reason maybe that the human action in UAV dataset is complicated, and the different shape of hand or leg should be paid different attention for classification. The experimental result with other methods is presented in the fourth column of Table 1, the proposed MSNet-bn-attention still achieves the best result.

5. Conclusion

In this paper, we successfully introduced a batch normalization technique for analyzing the distribution of submanifolds extracted by the multi-scale submanifold network. And then, we innovatively designed an attention mechanism specifically tailored to submanifolds mapped to Euclidean space. Experimental results on visual classification of gesture and human action recognition demonstrate the superiority of our approach. Compared to the recent MSNet, the proposed method usually has better performance on various datasets, especially on the human action recognition dataset of UAV. These results illustrate the signification of manifold normalization and the submanifolds attention model. To the best of our knowledge, this work is the first to integrate the multi-scale submanifold network with the submanifold normalization and submanifold attention mechanism.

Acknowledgments

This work was supported in part by the Natural Science Foundation of Guangdong Province under Grant 2021A1515011859 and 2020A1515010423, in part by the National

Natural Science Foundation of China under Grant 61701126,62176071, in part by the Key Laboratory of Philosophy and Social Sciences in Guangdong Province of Maritime Silk Road of Guangzhou University (GD22TWCXGC15), and in part by the Guangzhou Basic Research Program Jointly Funded by City and University under Grant 202102010395.

References

[1] Zhiwu Huang and Luc Van Gool. A Riemannian Network for SPD Matrix Learning. Proceedings of the AAAI Conference on Artificial Intelligence, 2017, 31(1). doi: 10.1609/aaai.v31i1.10866.

[2] Zhiwu Huang, Ruiping Wang, Shiguang Shan, Xianqiu Li, and Xilin Chen. Log-Euclidean Metric Learning on Symmetric Positive Definite Manifold with Application to Image Set Classification. Proceedings of International Conference on Machine Learning, 2015,37: 720–729.

[3] Zhiwu Huang, Jiqing Wu, and Luc Van Gool. Building Deep Networks on Grassmann Manifolds. Proceedings of the AAAI Conference on Artificial Intelligence, 2018,32 (1), doi: 10.1609/aaai.v32i1.11725.

[4] Rui Wang, Xiao-Jun Wu, and Josef Kittler. Graph Embedding Multi-Kernel Metric Learning for Image Set Classification With Grassmannian Manifold-Valued Features. IEEE Transactions on Multimedia, 2021,23: 228–242, doi: 10.1109/TMM.2020.2981189.

[5] Masoud Faraki, Mehrtash T. Harandi, and Fatih Porikli. A Comprehensive Look at Coding Techniques on Riemannian Manifolds. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29 (11): 5701–5712, doi: 10.1109/TNNLS.2018.2812799.

[6] Haoliang Sun, Xiantong Zhen, Yuanjie Zheng, Gongping Yang, Yilong Yin, and Shuo Li. Learning Deep Match Kernels for Image-Set Classification. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017:6240–6249,doi: 10.1109/CVPR.2017.661.

[7] Rui Wang, Xiao-Jun Wu, Zhen Liu, and Josef Kittler. Geometry-Aware Graph Embedding Projection Metric Learning for Image Set Classification. IEEE Transactions on Cognitive and Developmental Systems, 2022, 14 (3): 957–970, doi: 10.1109/TCDS.2021.3086814.

[8] Zhi Gao, Yuwei Wu, Mehrtash Harandi, and Yunde Jia. A Robust Distance Measure for Similarity-Based Classification on the SPD Manifold. IEEE Transactions on Neural Networks and Learning Systems, 2020,31 (9): 3230–3244, doi: 10.1109/TNNLS.2019.2939177.

[9] Xuan Son Nguyen, Luc Brun, Olivier Lezoray, and Sebastien Bougleux. A Neural Network Based on SPD Manifold Learning for Skeleton-Based Hand Gesture Recognition. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019:12028–12037, doi: 10.1109/CVPR.2019.01231.

[10] Rui Wang, Xiao-Jun Wu, Ziheng Chen, Tianyang Xu, and Josef Kittler. Learning a discriminative SPD manifold neural network for image set classification. Neural Networks, 2022,151: 94–110, doi: 10.1016/j.neunet.2022.03.012.

[11] Rui Wang, Xiao-Jun Wu, Ziheng Chen, Tianyang Xu, and Josef Kittler. DreamNet: A Deep Riemannian Manifold Network for SPD Matrix Learning. Asian Conference on Computer Vision, 2023,13846:646–663, doi: 10.1007/978-3-031-26351-4_39.

[12] Rui Wang, Xiao-Jun Wu, Tianyang Xu, Cong Hu, and Josef Kittler. U-SPDNet: An SPD manifold learning-based neural network for visual classification. Neural Networks, 2023,161: 382–396, doi: 10.1016/j.neunet.2022.11.030.

[13] Mehrtash Harandi, Mathieu Salzmann, and Richard Hartley. Dimensionality Reduction on SPD Manifolds: The Emergence of Geometry-Aware Methods. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018,40 (1): 48–62, doi: 10.1109/TPAMI.2017.2655048.

[14] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Projection Metric Learning on Grassmann Manifold with Application to Video based Face Recognition. IEEE Conference on Computer Vision and Pattern Recognition, 2015:140–149, doi: 10.1109/CVPR.2015.7298609.

[15] Ruiping Wang, Huimin Guo, L. S. Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In 2012 IEEE Conference on Computer Vision and Pattern Recognition,2012:2496–2503, doi: 10.1109/CVPR.2012.6247965.

[16] Rui Wang, Xiao-Jun Wu, Kai-Xuan Chen, and Josef Kittler. Multiple Manifolds Metric Learning with Application to Image Set Classification. International Conference on Pattern Recognition, 2018:627–632, doi: 10.1109/ICPR.2018.8546030.

[17] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, International conference on machine learning ,2015: 448-456.

[18] Daniel Brooks, Olivier Schwander, Frederic Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for SPD neural networks. Advances in Neural Information Processing Systems,2019:32.
[19] Tong Zhang, Wenming Zheng, Zhen Cui, Yuan Zong, Chaolong Li, Xiaoyan Zhou, and Jian Yang. Deep Manifold-to-Manifold Transforming Network for Skeleton-based Action Recognition. IEEE Transactions on Multimedia, 2020, doi: 10.1109/TMM.2020.2966878.
[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention,2015: 234–241.
[21] Ziheng Chen, Tianyang Xu, Xiao-Jun Wu, Rui Wang, Zhiwu Huang, and Josef Kittler. Riemannian Local Mechanism for SPD Neural Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 2023,37 (6): 7104–7112, doi: 10.1609/aaai.v37i6.25867.
[22] Qibin Hou, Daquan Zhou, and Jiashi Feng. Coordinate Attention for Efficient Mobile Network Design. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021:13708–13717, doi: 10.1109/CVPR46437.2021.01350.
[23] Or Yair, Mirela Ben-Chen, and Ronen Talmon. Parallel Transport on the Cone Manifold of SPD Matrices for Domain Adaptation. IEEE Transactions on Signal Processing, 2019,67 (7): 1797–1811, doi: 10.1109/TSP.2019.2894801.
[24] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Geometric Means in a Novel Vector Space Structure on Symmetric Positive-Definite Matrices. SIAM Journal on Matrix Analysis and Applications, 2007, 29 (1): 328–347, doi: 10.1137/050637996.
[25] Tae-Kyun Kim and R. Cipolla. Canonical Correlation Analysis of Video Volume Tensors for Action Categorization and Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31 (8): 1415–1428, doi: 10.1109/TPAMI.2008.167.
[26] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations.IEEE Conference on Computer Vision and Pattern Recognition, 2018:409–419, doi: 10.1109/CVPR.2018.00050.
[27] Tianjiao Li, Jun Liu, Wei Zhang, Yun Ni, Wenqian Wang, and Zhiheng Li. UAV-Human: A Large Benchmark for Human Behavior Understanding with Unmanned Aerial Vehicles. IEEE Conference on Computer Vision and Pattern Recognition,2021:16261–16270, doi: 10.1109/CVPR46437.2021.01600.
[28] Jihun Hamm and Daniel D Lee. Grassmann Discriminant Analysis: A Unifying View on Subspace-Based Learning, International Conference on Machine Learning,2008: 376-383.
[29] Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning. Pattern Recognition, 2015,48 (10): 3113–3124, doi: 10.1016/j.patcog.2015.03.011.
[30] Rui Wang, Xiao-Jun Wu, and Josef Kittler. SymNet: A Simple Symmetric Positive Definite Manifold Deep Learning Method for Image Set Classification. IEEE Transactions on Neural Networks and Learning Systems, 2022, 33 (5): 2208–2222, doi: 10.1109/TNNLS.2020.3044176.
[31] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. IEEE Conference on Computer Vision and Pattern Recognition, 2014:588–595, doi: 10.1109/CVPR.2014.82.
[32] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. IEEE Conference on Computer Vision and Pattern Recognition, 2015:1110–1118, doi: 10.1109/CVPR.2015.7298714.
[33] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly Learning Heterogeneous Features for RGB-D Activity Recognition. IEEE Conference on Computer Vision and Pattern Recognition,2015:5344-5352, doi: 10.1109/CVPR.2015.7299172.
[34] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. IEEE Conference on Computer Vision and Pattern Recognition, 2016:1933–1941, doi: 10.1109/CVPR.2016.213.
[35] Hossein Rahmani and Ajmal Mian. 3D Action Recognition from Novel Viewpoints. IEEE Conference on Computer Vision and Pattern Recognition, 2016:1506–1515, doi: 10.1109/CVPR.2016.167.
[36] Guillermo Garcia-Hernando and Tae-Kyun Kim. Transition Forests: Learning Discriminative Temporal Transitions for Action Recognition and Detection. IEEE Conference on Computer Vision and Pattern Recognition, 2017:407–415, doi: 10.1109/CVPR.2017.51.
[37] Tae Soo Kim and Austin Reiter. Interpretable 3D Human Action Analysis with Temporal Convolutional Networks. IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017:1623–1631, doi: 10.1109/CVPRW.2017.207.
[38] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified Egocentric Recognition of 3D Hand-Object Poses and Interactions. IEEE Conference on Computer Vision and Pattern Recognition, 2019:4506–4515, doi: 10.1109/CVPR.2019.00464.

[39] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable Architecture Search, International Conference on Learning Representations, 2019.
[40] Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. Fair DARTS: Eliminating Unfair Advantages in Differentiable Architecture Search, European Conference on Computer Vision, 2020, doi:10.1007/978-3-030-58555-6_28.
[41] Rui Wang, Xiao-Jun Wu, Kai-Xuan Chen, and Josef Kittler. Multiple Riemannian Manifold-Valued Descriptors Based Image Set Classification With Multi-Kernel Metric Learning. IEEE Transactions on Big Data,2022,8 (3): 753–769, doi: 10.1109/TBDATA.2020.2982146.
[42] Rudrasis Chakraborty, Jose Bouza, Jonathan H. Manton, and Baba C. Vemuri. ManifoldNet: A Deep Neural Network for Manifold-Valued Data With Applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022,44 (2): 799–810, doi: 10.1109/TPAMI.2020.3003846.
[43] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Training Deep Networks with Structured Layers by Matrix Backpropagation, International Conference on Computer Vision, 2015.