SymNet: A Simple Symmetric Positive Definite Manifold Deep Learning Method for Image Set Classification

Rui Wang, Xiao-Jun Wu, and Josef Kittler, *Life Member, IEEE*

***Abstract*— By representing each image set as a nonsingular covariance matrix on the symmetric positive definite (SPD) manifold, visual classification with image sets has attracted much attention. Despite the success made so far, the issue of large within-class variability of representations still remains a key challenge. Recently, several SPD matrix learning methods have been proposed to assuage this problem by directly constructing an embedding mapping from the original SPD manifold to a lower dimensional one. The advantage of this type of approach is that it cannot only implement discriminative feature selection but also preserve the Riemannian geometrical structure of the original data manifold. Inspired by this fact, we propose a simple SPD manifold deep learning network (SymNet) for image set classification in this article. Specifically, we first design SPD matrix mapping layers to map the input SPD matrices into new ones with lower dimensionality. Then, rectifying layers are devised to activate the input matrices for the purpose of forming a valid SPD manifold, chiefly to inject nonlinearity for SPD matrix learning with two nonlinear functions. Afterward, we introduce pooling layers to further compress the input SPD matrices, and the log-map layer is finally exploited to embed the resulting SPD matrices into the tangent space via log-Euclidean Riemannian computing, such that the Euclidean learning applies. For SymNet, the (2-D)^{2}principal component analysis (PCA) technique is utilized to learn the multistage connection weights without requiring complicated computations, thus making it be built and trained easier. On the tail of SymNet, the kernel discriminant analysis (KDA) algorithm is coupled with the output vectorized feature representations to perform discriminative subspace learning. Extensive experiments and comparisons with state-of-the-art methods on six typical visual classification tasks demonstrate the feasibility and validity of the proposed SymNet.**

***Index Terms*— Deep learning, image set classification, nonsingular covariance matrix, symmetric positive definite (SPD) manifold, (2-D)^{2}principal component analysis (PCA).**

I. Introduction

With the rapid development of multimedia technologies, a huge number of videos have been recorded in the community of computer vision and pattern recognition (CV&PR). As each video sequence can be treated as an image set, image set classification has been attracting growing attention [1]–[11]. Video-based face recognition [4], [5], [9], [10], video-based face verification [7], [13], video-based facial emotion recognition [12], dynamic scene classification [10], [11], [14], and action recognition [12], [15] are some of its practical applications. Different from the traditional visual classification problem where the decision-making is based on a single still image, for image set classification, both the gallery and probe samples are image sets, each of which makes up a number of images belonging to the same category. Another distinguishing feature of image set is that it can provide more data variability information for classification. However, video data usually involve a wide range of within-class variations, caused by illumination, pose, expression, and changes of other conditions in the video capturing process. Therefore, how to properly encode such variational information and learn invariant representations is considered as a pivotal challenge.

Among the existing set models, covariance matrix has gained remarkable success in image set characterization. Its main advantage is the simplicity, flexibility, and sufficiency in describing each video clip with a different number of frames as a fixed-dimensional second-order representation [9], [16]. Therefore, we choose it as the feature descriptor for set data in this article. As well studied in [4] and [17], the underlying space of a family of nonsingular covariance matrices with the same dimensionality is usually considered not to be a vector space, but instead adhering to a type of nonlinear Riemannian manifold, i.e., symmetric positive definite (SPD) manifold. Accordingly, the conventional Euclidean computations cannot be applied to the SPD manifold-valued data directly. To address this issue, Arsigny *et al.* [18], Pennec *et al.* [17], and Sra [19] advocated some Riemannian metrics for similarity measurement between SPD matrices, such as affine-invariant Riemannian metric (AIRM) [18] and log-Euclidean metric (LEM) [17]. By utilizing these well-studied Riemannian metrics, several SPD matrix learning methods [7], [21], [22] are suggested to transform the SPD manifold-valued data into tangent space representations such that the Euclidean computations apply. Alternatively, Wang *et al.* [4] and Harandi *et al.* [23] proposed to exploit the Riemannian

Manuscript received January 9, 2020; revised July 22, 2020; accepted November 3, 2020. Date of publication March 30, 2021; date of current version May 3, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62020106012, Grant U1836218, and Grant 61672265; in part by the 111 Project of Ministry of Education of China under Grant B12018; and in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/N007743/1 and Grant MURI/EPSRC/DSTL EP/R018456/1. *(Corresponding author: Xiao-Jun Wu.)*

Rui Wang and Xiao-Jun Wu are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China, and also with the Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China (e-mail: cs_wr@jiangnan.edu.cn; xiaojun_wu_jnu@163.com).

Josef Kittler is with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: j.kittler@surrey.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2020.3044176.

Digital Object Identifier 10.1109/TNNLS.2020.3044176

kernel functions to embed the original SPD manifold into a reproducing kernel Hilbert space (RKHS) for subsequent Euclidean learning. However, the approaches mentioned earlier typically convey the idea of approximate computation. In consequence, the Riemannian geometrical structure of the original data manifold cannot be fully exploited by the feature transformation process.

To tackle this problem, several SPD manifold discriminant analysis methods have been put forward to keep an eye on geometry-aware feature embedding and selection [7], [9], [12], [22]. The working mechanism of them is to jointly perform embedding mapping learning and similarity metric learning for the original SPD manifold-valued data. As a result, a lower dimensional and more appropriate SPD manifold can be yielded. Nevertheless, for some sophisticated visual classification tasks, the aforementioned learning algorithms still remain a research gap to mine more powerful semantic representations [7] for visual data. To remedy this research gap, some researchers attempt to model each image set simultaneously with its first-, second-, and high-order statistics, in view of their complementarity in encoding the geometrical structure of set data [5], [24]–[26]. Considering that different statistics lie in different topological spaces, the Riemannian kernel functions are first applied to explicitly embed these heterogeneous features into kernel Hilbert spaces. Then, the metric learning framework is utilized to merge these hybrid kernel features into a compact and efficient subspace for classification.

In the past decade, deep learning technique [27], [28] has gradually become a vital tool for learning desirable, reliable, and powerful feature representations in the CV&PR community. By extending the Euclidean network paradigm to the SPD manifold, several SPD manifold deep learning networks have been established to open up a new orientation for SPD matrix nonlinear learning [12], [15]. These networks consist of a stack of trainable blocks, followed by a normally used fully connected layer and a supervised classifier. In each block, the feature learning process is implemented by an SPD matrix mapping layer (similar to 2-D convolutional layer) and a nonlinear rectifying layer. Compared with the traditional SPD manifold learning methods, this type of approach further lifts the classification performance on some challenging visual scenarios. The reasons for its success arise from two innovations: 1) the end-to-end architecture generalizes the conventional SPD matrix learning to deep and nonlinear functions and 2) Riemannian matrix backpropagation optimizer. To the best of our knowledge, the study of deep learning in the context of Riemannian manifolds is still in its infancy, how to design a universal network for a variety of computer vision problems and how to seek proper training strategies are considered as two major challenges.

In this article, we plan to design a simple SPD matrix learning network with the following two characteristics: 1) compared with some representative image set classification methods, its classification performance is competitive and 2) its computational efficiency has a considerable improvement. To cope with this objective, the SPD matrix mapping layer is first designed to generate new representations from the input SPD matrices by exploiting the $(2\text{-D})^2$principal component analysis (PCA) technique to perform unsupervised filter learning. Then, the rectifying layer is devised to introduce a nonlinear learning scheme by regularizing the input matrices with two nonlinear functions. To further compress the extracted geometric features, we generalize the conventional pooling operation (e.g., max or mean) to the proposed SymNet in three steps: 1) utilizing the matrix logarithm map to embed them into a tangent space; 2) conducting pooling operation in this space; and 3) exploiting the matrix exponential map to map these pooled data back into the SPD manifold. Hence, we name it the tangent space pooling strategy. In addition, in this article, we also study another two pooling tactics for the proposed model: 1) directly performing the conventional max-pooling operation on the SPD manifold and 2) the Fréchet mean-based SPD manifold mean pooling. With the help of a stack of SPD matrix mapping, rectifying, and pooling layers, the input SPD matrices could be transformed into some lower dimensional and more efficient counterparts. Due to the Euclidean classifiers that cannot be directly applied to them for image set classification, the log-map layer is finally exploited to convert these SPD manifold-valued data learned by SymNet into Euclidean representations by utilizing the log-Euclidean Riemannian computation. In what follows, the kernel discriminant analysis (KDA) algorithm is utilized to carry out discriminant subspace learning. Accordingly, the main differences between the proposed model and most existing deep learning methods (e.g., [29]–[31], [37]) are twofold: 1) both the inputs and outputs of our network are the structured SPD matrices, which means that this network is strictly constructed in the scenario of SPD manifold; 2) different from the sophisticated backpropagation optimizer-based end-to-end training, the shallow learning algorithm (i.e., $(2\text{-D})^2$PCA technique) is adopted to the newly designed lightweight cascaded architecture for training. In this article, our main contributions can be summarized as follows.

1) We develop a lightweight cascaded network for SPD matrix nonlinear learning.
2) To make the proposed SymNet be built and trained easier, we make use of the $(2\text{-D})^2$PCA algorithm to perform unsupervised filter learning rather than the Riemannian matrix backpropagation computing-based end-to-end training.
3) We design the rectifying layer with two nonlinear activation functions to endow the proposed model with nonlinear learning mechanism, chiefly to alleviate the intrasubject ambiguity of representations.
4) We study three different pooling strategies and demonstrate that the tangent space pooling method is more suitable for our SymNet.

II. Related Work

To the best of our knowledge, the existing SPD manifold-based image set classification methods can be grouped into four categories, i.e., the kernel-based methods, SPD manifold dimensionality reduction (DR) methods, multiple statistics fusion-based methods, and SPD manifold deep learning methods. They are reviewed as follows.

A. Kernel-Based Methods

For this kind of approach [1], [4], [10], [32]–[35], the original SPD manifold is transformed into an explicit kernel space via the well-studied Riemannian kernel functions. As a result, the Euclidean computations can be applied to carry out feature learning and classification. Therein, Wang *et al.* [4] and Vemulapalli *et al.* [33] exploited the LEM derived Riemannian kernel function to map the data from SPD manifold to RKHS, followed by the Euclidean classifiers for image set classification. Wang *et al.* [1] investigated to encode the SPD manifold

of Gaussians in the mapped kernel space by deriving a series of probabilistic kernels. To improve the description ability of the existing Stein kernel-based methods, Zhang *et al.* [34] devised a discriminative Stein kernel for SPD matrices representation learning and similarity measurement. Harandi *et al.* [23], [35] generalized the conventional sparse coding and dictionary learning to the SPD manifold by using the Stein kernel to embed the SPD manifold-valued data into RKHS.

B. SPD Manifold DR Methods

To overcome an obvious limitation of the kernel-based methods (i.e., the Riemannian geometry of the original SPD manifold is distorted by the process of Hilbert space embedding), Harandi *et al.* [9] tried to produce a lower dimensional and more powerful new SPD manifold via an orthogonal mapping obtained by performing similarity metric learning in the original feature space. Similarly, Huang *et al.* [7] designed an LEM learning framework to directly transform the original tangent space into a more compact one, where the SPD manifold properties are preserved. Recently, Zhou *et al.* [22] developed a new version of LEM learning (LEML) [7] named α-covariance-like metric learning (CML), which aims to learn a sample-specific transformation matrix rather than the fixed one in LEML. Consequently, the SPD manifold-valued feature representations learned by α-CML will exhibit more discriminatory power.

C. Multiple Statistics Fusion-Based Methods

Different from the aforementioned learning approaches that encode the image set data only using the second-order statistical descriptor, Lu *et al.* [24] performed set data modeling simultaneously with the first-, second-, and third-order (tensor) statistics for the sake of extracting complementary structural information. To make better use of these heterogeneous features, a multikernel metric learning framework is proposed to fuse them into a discriminative unified subspace for classification. However, this method applies the Euclidean kernel function to perform RKHS embedding, which is unable to veritably reflect the geometrical structure of the original higher order statistics in the new feature space. Taking this into account, Huang *et al.* [26], Wang *et al.* [25], and Wang *et al.* [5] endowed different statistics of each image set with different kernel functions for explicit kernel spaces transformation. Afterward, the metric learning framework is designed for hybrid features fusion and classification.

D. Riemannian Manifold Deep Learning Methods

Inspired by the proven effectiveness of ConvNets in learning powerful feature representations, Sun *et al.* [14] aggregated the local match kernels built upon arc-cosine similarity with a deep neural architecture to form a global match kernel for more discriminative similarity measurement. Lu *et al.* [36] proposed to integrate manifold metric leaning into convolutional neural network (CNN) for the purpose of extracting fine-grained and class-specific information for improved classification. More recently, some researchers extend the ideology of deep learning to Riemannian manifolds for the sake of mining more fine-grained geometric features of visual data. In this scenario, a slice of Riemannian deep neural networks have been devised, such as SPDNet [12], which is made up of a stack of SPD matrix transformation and nonlinear activation layers to learn hierarchical and structured semantic representations. Inspired by Huang and Gool [12], Nguyen *et al.* [15] designed a novel deep neural network on the Gaussian embedded Riemannian manifold for 3-D hand gesture recognition in the spatial and temporal domains. The well documented Riemannian deep neural networks also include DeepO2P [37], which is built to embed the global structured computations into deep architecture for semantic segmentation, GCNN [38], a generalized CNN framework for hierarchically learning task-specific feature representations, and GrNet [39], which performs deep learning in the context of Grassmannian manifold.

TABLE I
LIST OF SYMBOLS USED AND THEIR CORRESPONDING EXPLANATIONS

Symbols	Explanation
m_k	the number of feature maps of the k-th layer
$\mathbb{K}$	the number of layers of this network
r, k	$r = 1, 2, ..., m_k$, $k = 1, 2, ..., \mathbb{K}$
W_k^r	the r-th connection weight of the k-th layer
X_k	the output SPD matrix of the k-th layer
N	the number of image sets of the gallery $\mathcal{T}$
v	non-zero vector in $\mathbb{R}^d$
S_i	the i-th training image set, where $S_i \in \mathbb{R}^{d\times n_i}$
n_i	the number of frames contained in S_i
s_i^j	the j-th frame of S_i, where $s_i^j \in \mathbb{R}^{d\times 1}$
μ_i	the mean of S_i, where $\mu_i \in \mathbb{R}^{d\times 1}$
X^i	the i-th covariance matrix computed by S_i
Sym_d^+	the space of SPD real $d \times d$ matrices
f_m	SPD matrix mapping layer
f_r	SPD matrix rectifying layer
f_p	SPD matrix pooling layer
$f_{\log}$	SPD matrix log-map layer

III. PROPOSED ALGORITHM

Fig. 1 shows an overview of the proposed lightweight cascaded network for SPD matrix learning. The different components of our model are detailedly presented in Sections III-A–III-E. In Section III-F, we give a brief introduction to the KDA algorithm, which is used to train a discriminative SymNet. Finally, Section III-G analyzes the relationship between the proposed network and SPDNet [12]. The used symbols in this article and their corresponding explanations are listed in Table I.

A. SPD Matrix Mapping Layer

The most important target of the proposed SymNet is to produce more compact and efficient feature matrices. To preserve the Riemannian geometrical structure of the input SPD matrices in new feature space, each resulting matrix should satisfy symmetric positive definiteness. With this objective, we design this SPD matrix mapping layer to transform the input SPD matrices that reside on $\text{Sym}_{d_{k-1}}^+$ into some new ones that lie in $\text{Sym}_{d_k}^+$ via a bilinear mapping f_m, formulated as

$$X_k = f_m(W_k, X_{k-1}) = W_k^T X_{k-1} W_k \tag{1}$$

where $X_{k-1} \in \mathbb{R}^{d_{k-1}\times d_{k-1}}$ is the input SPD matrix of kth layer, $X_k \in \mathbb{R}^{d_k\times d_k}$ is the resulting SPD matrix, and $W_k \in \mathbb{R}^{d_{k-1}\times d_k}$ ($d_k \le d_{k-1}$) is the to-be-learned transformation matrix (connection weights). In this article, we utilize a (2-D)^{2}PCA algorithm to perform unsupervised parameter learning. Now, we first give the definition of the SPD manifold and then introduce the (2-D)^{2}PCA algorithm.

The Definition of SPD Manifold: A $d \times d$ symmetric real matrix X is said to be positive definite if $v^T X v > 0$ for all nonzero v in $\mathbb{R}^d$, and the SPD manifold represented by Sym_d^+ is spanned by a set of $d \times d$ SPD matrices

$$\text{Sym}_d^+ = \{X \in \mathbb{R}^{d\times d} : v^T X v > 0 \quad \forall v \in \hat{\mathbb{R}}^d\} \tag{2}$$

where $\hat{\mathbb{R}}^d$ is $\mathbb{R}^d$ space without the zero vector.

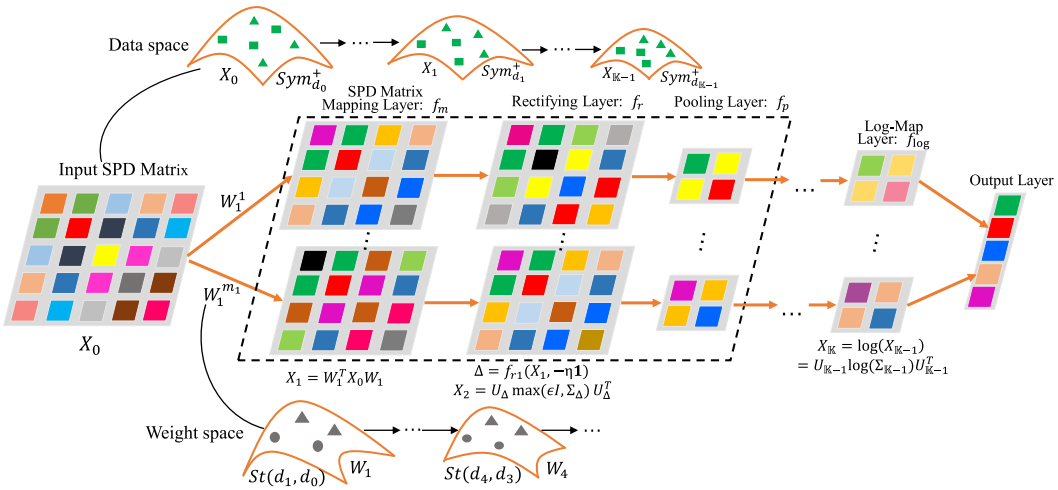


Fig. 1. Schematic of the proposed SymNet framework. It is mainly made up of the SPD matrix mapping layers to generate lower dimensional and more appropriate SPD matrices via learnable mapping W_k, rectifying layers to introduce nonlinear learning mechanism for SPD matrices, SPD matrix pooling layer to further compress the learned SPD manifold-valued features, and log-map layer to perform Riemannian computing. The upper part of this figure indicates that our model is strictly constructed on the SPD manifold $\mathrm{Sym}_{d_k}^+$, spanned by a set of d_k-dimensional SPD matrices. The lower part of this figure demonstrates that the weight space of each SPD matrix mapping layer is a compact Stiefel manifold $St(d_k, d_{k-1})$.

(2-D)²PCA Algorithm: Let $T = [S_1, S_2, \ldots, S_N]$ be the gallery consists of N image sets and $L = [l_1, l_2, \ldots, l_N] \in \mathbb{R}^{1\times N}$ be the corresponding label vector. With these notations, the ith covariance matrix that corresponds to S_i can be computed as

$$X^i = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}\left(s_i^j - \mu_i\right)\left(s_i^j - \mu_i\right)^T. \tag{3}$$

To make the (2-D)²PCA algorithm play a role, we first treat each X^i as a basic sample. Then, the sample covariance matrix computed from the row direction is expressed as

$$\mathbb{C} = \frac{1}{N-1}\sum_{i=1}^{N}(X^i - \bar{X})^T(X^i - \bar{X}) \tag{4}$$

where $\bar{X}$ is the mean of all the basic samples computed by (3). Similarly, it can also be described from the perspective of column direction as

$$\mathbb{C}_c = \frac{1}{N-1}\sum_{i=1}^{N}(X^i - \bar{X})(X^i - \bar{X})^T. \tag{5}$$

It is easy to check that (4) equals (5). For simplicity, we take $\mathbb{C}$ to perform subsequent computations. Next, (2-D)²PCA tries to minimize the following reconstruction error to learn the target transformation matrix:

$$\min_{\mathcal{M}\in\mathbb{R}^{d_{k-1}\times\mathcal{P}}}\sum_{i=1}^{N}\|X^i - \mathcal{M}\mathcal{M}^T X^i\|_{\mathrm{F}}^2, \quad \text{s.t. } \mathcal{M}^T\mathcal{M} = I_{\mathcal{P}} \tag{6}$$

where $I_{\mathcal{P}}$ is an identity matrix of size $\mathcal{P}\times\mathcal{P}$. In fact, (6) is an eigenvalue problem and its solution is composed of a family of eigenvectors corresponding to the $\mathcal{P}$ largest eigenvalues of $\mathbb{C}$. Due to the produced SPD matrices of the kth layer that should be of the same size, each connection weight W_k^r can be described as

$$W_k^r = \mathrm{div}_{d_{k-1},d_k}(V(\mathbb{C})) \in \mathbb{R}^{d_{k-1}\times d_k}, \quad r = 1, 2, \ldots, m_k \tag{7}$$

where $V(\mathbb{C})$ represents a matrix composed by $\mathcal{P}$ ($\mathcal{P} = d_k \times m_k$) leading eigenvectors of $\mathbb{C}$, and $\mathrm{div}_{d_{k-1},d_k}(V)$ is a function that can successively divide $V(\mathbb{C})$ into m_k nonoverlapping parts with each part making up of d_k eigenvectors.

Having obtained these transformation matrices, the new SPD matrix X_k with respect to the input one can be generated. According to Theorem 1, W_k is required to satisfy the column full rank to ensure that X_k lies on a valid SPD manifold.

Theorem 1: Given an SPD matrix $X \in \mathbb{R}^{d_1\times d_1}$ and an projection matrix $W \in \mathbb{R}^{d_1\times d_2}$, $d_2 \le d_1$. Let $X' = W^T X W$, we say that X' is an SPD matrix of size $d_2 \times d_2$ if and only if W is a column full rank matrix, i.e., $rank(W) = d_2$.

Proof: 1) Assume that X is an SPD matrix and W satisfies column full rank. For any nonzero vector $\upsilon \in \mathbb{R}^{d_2}$, we can get the following equation:

$$\upsilon^T X'\upsilon = \upsilon^T W^T X W\upsilon = (W\upsilon)^T X(W\upsilon). \tag{8}$$

Because W is a column full rank matrix and υ is a nonzero vector, we can get $W\upsilon \neq 0$, which leads to $\upsilon^T X'\upsilon > 0$. According to the definition of the SPD manifold, X' is an SPD matrix, which is proved.

2) Assume that X' is an SPD matrix. For any nonzero vector $\upsilon \in \mathbb{R}^{d_2}$, we can easily have $\upsilon^T X'\upsilon > 0$. Based on (8), it is equivalent to $(W\upsilon)^T X(W\upsilon) > 0$. Due to X that is an SPD matrix, we can get $W\upsilon \neq 0$. Obviously, $rank(W) = d_2$, and W is a column full rank matrix.

B. Rectifying Layer

Rectified linear unit (ReLU) is well known for its efficacy in improving the discriminative performance of ConvNets [27], [28] by rectifying the undesired results. Accordingly, generalizing this paradigm to the domain of Riemannian deep learning and devising similar operations for the Riemannian manifold-valued data may also be indispensable. To achieve this objective, we design the rectifying layer to regularize the input SPD matrices for the sake of introducing nonlinear feature learning mechanism using two nonlinear activation functions. The first activation function $\Delta = f_{r_1}(X_{k-1}, -\eta\mathbf{1})$ is formulated as

$$\Delta(i,j) = \begin{cases} -\eta, & \text{if } i \neq j \text{ and } X_{k-1}(i,j) \in (-\eta, 0] \\ X_{k-1}(i,j), & \text{otherwise} \end{cases} \tag{9}$$

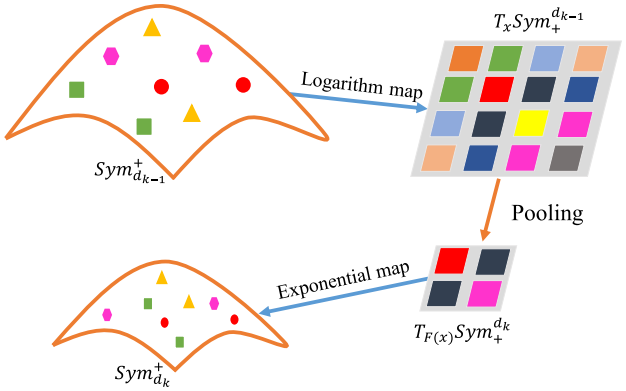


Fig. 2. Illustration of the SPD matrix pooling process.

where $\mathbf{1} \in \mathbb{R}^{d_{k-1} \times d_{k-1}}$ is a matrix of all ones and η is an activation threshold. From (9), we can find that some negative correlation values of each input SPD matrix are amplified toward the negative direction, which is able to mitigate the intrasubject variations to a certain extent.

The second activation function is mainly designed to tune up the small eigenvalues of each generated Δ such that its positive definiteness can be ensured. The specific form of this function, inspired by $\max(0, x)$, is presented as follows [12]:

$$X_k = f_{r_2}(\Delta) = U \max(\epsilon I, \Sigma) U^T \tag{10}$$

where U and Σ are, respectively, the eigenvector and eigenvalue matrices, obtained by applying the singular value decomposition (SVD) to Δ, i.e., $\Delta = U \Sigma U^T$, ϵ is a small rectification threshold, and $\max(\epsilon I, \Sigma)$ is a diagonal matrix defined as

$$\max(\epsilon I, \Sigma)_{ii} = \begin{cases} \Sigma_{ii}, & \text{if } \Sigma_{ii} > \epsilon \\ \epsilon, & \text{otherwise.} \end{cases} \tag{11}$$

The operations studied in this layer are the core nonlinear embedding mechanisms of the proposed SymNet.

C. SPD Matrix Pooling Layer

In the field of ConvNets, pooling operations are often used to induce robustness to registration errors and to reduce the number of parameters [27], [28], [40]. In this sense, introducing similar operations in the context of Riemannian deep learning may also be interesting. As an exploration, we propose to pool the input SPD matrices by first utilizing the logarithm map to map them into an associated tangent space such that the conventional pooling operation (e.g., max or mean) applies. In what follows is the exponential map, exploited for the retraction of these refined data back into the SPD manifold. For more detailed treatment about the two maps, please kindly refer to [17] and [41]. Fig. 2 shows an intuitive illustration of the introduced tangent space pooling strategy. Here, the patch size we set in this article is 2×2, which means that the redundancy of the input SPD matrices could be reduced up to a point after using this pooling method.

In addition to the tangent space pooling tactic mentioned earlier, we note that directly performing max pooling on the SPD manifold can also realize the purpose of DR and Riemannian geometrical structure retention for the input SPD matrices. Due to the working mechanism of conventional mean pooling operation that is to compute the arithmetic mean of the features within a target region, it cannot be applied to the SPD manifold-valued data because of the destruction of the Riemannian geometrical structure. Despite this, this issue could be tackled in two stages from the perspective of DR.

To be specific, at the first stage, we need to compute the mean of a set of input SPD matrices of the kth layer $\{{}^i X_{k-1}^r\}_{i=1}^N$. Since these points reside on the SPD manifold, we make use of the Fréchet formulation instead of the arithmetic mean for computation. This can be described as

$$P^* = \arg\min_{P \in \mathrm{Sym}_{d_{k-1}}^+} \sum_{i=1}^{N} \sum_{r=1}^{m_k} \mathcal{D}_{\mathrm{LEM}}^2 \left({}^i X_{k-1}^r, P\right) \tag{12}$$

where $\mathcal{D}_{\mathrm{LEM}}^2({}^i X_{k-1}^r, P) = ||\log({}^i X_{k-1}^r) - \log(P)||_{\mathrm{F}}^2$ is the widely used LEM [17] for SPD matrices comparing, P is the mean of $\{{}^i X_{k-1}^r\}_{i=1}^N$, and $\log(\cdot)$ represents the matrix principal logarithm.

Theorem 2: The Fréchet mean of a set of SPD matrices $\{{}^i X_{k-1}^r\}_{i=1}^N$ with respect to $\mathcal{D}_{\mathrm{LEM}}$ is

$$P^* = \exp\left[\frac{1}{N m_k} \sum_{i=1}^{N} \sum_{r=1}^{m_k} \log\left({}^i X_{k-1}^r\right)\right]. \tag{13}$$

Proof: According to (12), the Fréchet mean must satisfy

$$\frac{\partial \sum_{i=1}^{N} \sum_{r=1}^{m_k} \mathcal{D}_{\mathrm{LEM}}^2 \left({}^i X_{k-1}^r, P\right)}{\partial P} = 0. \tag{14}$$

Given that

$$\frac{\partial \mathcal{D}_{\mathrm{LEM}}^2 \left({}^i X_{k-1}^r, P\right)}{\partial P} = -2P^{-1}[\log\left({}^i X_{k-1}^r\right) - \log(P)] \tag{15}$$

the result presented in (13) can be obtained.

Inspired by Harandi *et al.* [9], the goal of mapping the high-dimensional SPD manifold to a lower dimensional one can be achieved by solving the following optimization problem at the second stage:

$$\begin{aligned} M_k^* = & \arg\max_{M_k} \sum_{i=1}^{N} \sum_{r=1}^{m_k} \mathcal{D}_{\mathrm{LEM}}^2 \left(M_k^T \left({}^i X_{k-1}^r\right) M_k, M_k^T P M_k\right) \\ & \text{s.t. } M_k^T M_k = I_{d_k} \end{aligned} \tag{16}$$

where $M_k \in \mathbb{R}^{d_{k-1} \times d_k}$ is the to-be-learned embedding mapping of the kth layer. As discussed in [9] and [41], (16) corresponds to an optimization problem on the Grassmann manifold $\mathcal{G}(d_k, d_{k-1})$ and can be solved by exploiting the Grassmannian conjugate gradient (CG) method. Accordingly, we need to first compute the gradient of $\sum_{i=1}^{N} \sum_{r=1}^{m_k} \mathcal{D}_{\mathrm{LEM}}^2(\cdot, \cdot)$ with respect to M_k, which is given as

$$\begin{aligned} & \nabla_{M_k} \left[\sum_{i=1}^{N} \sum_{r=1}^{m_k} \mathcal{D}_{\mathrm{LEM}}^2 \left(M_k^T \left({}^i X_{k-1}^r\right) M_k, M_k^T P M_k\right)\right] \\ & = D_{M_k} \mathrm{Tr}\left[M_k^T \left(\sum_{i=1}^{N} \sum_{r=1}^{m_k} \Theta M_k M_k^T \Theta\right) M_k\right] \\ & = 4 \sum_{i=1}^{N} \sum_{r=1}^{m_k} \Theta M_k M_k^T \Theta M_k \end{aligned} \tag{17}$$

where $\Theta = \log({}^i X_{k-1}^r) - \log(P)$. With this gradient, the subsequent computations for learning M_k can be activated. Please kindly refer to Section 3.3 of [41] and [9] for more detailed introduction to CG on the Grassmann manifold.

D. Log-Map Layer

As well studied in [17], the LEM is capable of forming a Lie group structure for the SPD matrices such that the

SPD manifold can be embedded into a flat space under the matrix logarithm operator logm(·). Due to the Euclidean computations that are applicable to the domain of SPD matrix logarithms, the log-Euclidean Riemannian computation [17] is applied to the input SPD matrices in the $\mathbb{K}$th layer, which is expressed as

$$\begin{aligned} X_{\mathbb{K}} &= f_{\log}(X_{\mathbb{K}-1}) = \log(X_{\mathbb{K}-1}) \\ &= U_{\mathbb{K}-1}\mathrm{diag}(\log(\Sigma_{\mathbb{K}-1}))U_{\mathbb{K}-1}^T \end{aligned} \tag{18}$$

where $X_{\mathbb{K}-1} = U_{\mathbb{K}-1}\mathrm{diag}(\Sigma_{\mathbb{K}-1})U_{\mathbb{K}-1}^T$ is obtained by adopting eigenvalue decomposition to $X_{\mathbb{K}-1}$ and $\mathrm{diag}(\log(\Sigma_{\mathbb{K}-1}))$ is a diagonal matrix making up of the eigenvalue logarithms.

E. Output Layer

In order to facilitate the subsequent computations, we first vectorize the rth symmetric matrix ${}^iX_{\mathbb{K}}^r$ of the $\mathbb{K}$th (log-map) layer and then splice all the $m_{\mathbb{K}}$ vectors into a complete vector representation $V_i \in \mathbb{R}^{m_{\mathbb{K}}d_{\mathbb{K}}^2 \times 1}$ corresponding to the original ith ($i = 1 \rightarrow N$) image set S_i. This means that the original training samples have new geometric representations, with the aid of SymNet

$$[S_1, S_2, \ldots, S_N] \overset{\text{SymNet}}{\longrightarrow} [V_1, V_2, \ldots, V_N]. \tag{19}$$

In the test phase, each query set also needs to be processed by the same way.

F. Learning With KDA

KDA is well known for its effectiveness in learning an appropriate subspace for classification. Its working mechanism is to first transform each input sample V_i ($i = 1 \rightarrow N$) from vector space to RKHS via a feature map $\phi : \mathbb{R}^{m_{\mathbb{K}}d_{\mathbb{K}}^2} \mapsto \mathcal{H}$, $V_i \rightarrow \phi(V_i)$. Therefore, an inner product can be formulated in the mapped space $\mathcal{H}$ as $\langle \phi(V_i), \phi(V_j) \rangle = k(V_i.V_j)$. Then, a lower dimensional and more discriminative subspace can be generated under an embedding mapping learned by solving the following optimization problem [42]:

$$\alpha_{\text{opt}} = \arg\max \frac{\alpha^T K U K \alpha}{\alpha^T K K \alpha} \tag{20}$$

where α is the target transformation matrix, K is the kernel Gram matrix: $K_{ij} = k(V_i, V_j)$, and U is the block diagonal matrix expressed as

$$U_{ij} = \begin{cases} 1/n_k, & \text{if } V_i \text{ and } V_j \text{ come from the } k\text{th class} \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

Here, n_k indicates the number of samples used to train in the kth class. Due to the optimization problem in (20) that is equivalent to the following eigenvalue problem: $KUK\alpha = \lambda KK\alpha$, the optimal α is comprised of a set of eigenvectors corresponding to $(c-1)$ largest eigenvalues. For the learned subspace features, the nearest neighbor (NN) classifier is used for image set classification.

G. Relation With the Previous Works

Our method is closely related to [12]. Here, we point out some essential differences between the proposed SymNet and those introduced in [12].

First, SymNet utilizes the simple but efficient (2-D)^2 PCA [43] algorithm to conduct unsupervised filter learning, whereas SPDNet [12] exploits the Riemannian matrix back-propagation computing to perform end-to-end training, which is more time consuming than ours. Second, on the tail of the network, SPDNet makes use of the classical fully connected layer to learn Euclidean feature representations. Instead, SymNet utilizes KDA to perform discriminative subspace learning. As a result, training SymNet is very easy. Third, SPDNet introduces the nonlinear mapping scheme by tuning up some small eigenvalues with a ReLU-like function in the designed ReEig layer. However, the proposed SymNet additionally considers the impact of some negative elements of the SPD matrices on the discriminability of the learned features and designs a nonlinear activation function in the rectifying layer to adjust them to desired ones. Finally, the conventional pooling operations are generalized to SymNet to further compress the learned SPD matrices. However, SPDNet [12] does not take this into consideration.

IV. Experiments

To evaluate the effectiveness of the proposed SymNet,[1] we apply it to five different visual classification tasks: video-based face recognition, set-based object categorization, set-based cell identification, dynamic scene classification, and video-based facial emotion recognition. For the task of face recognition, the widely used YouTube celebrities (YTC) [4], [9] data set is our choice. As to the object categorization task, we utilize the ETH-80 [2] data set. The Virus data set [44] is applied to the task of cell identification. For the tasks of dynamic scene classification and facial emotion recognition, we exploit the Modeling Dynamic Scenes Dataset (MDSD) [45] and AFEW [12], [46] data sets, respectively.

A. Comparative Methods and Settings

We compare the proposed SymNet with the following image set classification methods: AIRM [18], Stein divergence [19], LEM [17], covariance discriminant learning (CDL) [4], Grassmann discriminant analysis (GDA) [2], Grassmannian graph-embedding discriminant analysis(GEDA) [48], localized multikernel metric learning (LMKML) [24], projection metric learning (PML) [13], LEML [7], SPD Manifold Learning based on Stein divergence (SPDML-Stein) and AIM (SPDML-AIM) [9], and SPD Manifold Network (SPDNet) [12].

We should point out that the classification results of these comparative methods on the five used data sets are obtained by running the source codes provided by the original authors, except for LMKML. Since its source code has not been released, we carefully reimplement it by referring to [24]. For a fair comparison, the parameters that we set in this article are empirically tuned according to the original works. For CDL, the perturbation is set to $10^{-3} \times trace(C)$. For GDA and GEDA, the number of basis vectors used to form the subspace is determined by cross validation. In PML, the dimensionality of the target Grassmann manifold and the value of the tradeoff coefficient α are set according to the original work [13]. For LEML, the values of η and ζ are chosen in the scope of $[0.1, 1, 10]$ and $[0.1 : 0.1 : 1]$, respectively. In SPDNet, the sizes of the transformation matrices are configured as 400×200, 200×100, and 100×50. Other parameters, such as the learning rate and the batch size, are searched by cross validation. For SPDML-Stein and SPDML-AIM, the two graph parameters υ_w and υ_b are determined according to the original work [9]. Note that for the parameters determined by cross validation, we report the best classification results for such methods in this article.

[1]The source code has been released on: https://github.com/GitWR/SymNet

TABLE II
SUITABLE VALUES FOR SymNet-v1 PARAMETERS

Dataset	ETH-80	YTC	Virus	AFEW	MDSD
d_m^{v1}	20	60	30	70	40
m^{v1}	8	4	12	5	4
η_r^{v1}	5.0	0.5	30.5	6E-5	2.98
ϵ_r^{v1}	1E-3	5E-5	1E-3	1E-3	1E-3

TABLE III
SUITABLE VALUES FOR SymNet-v2 PARAMETERS

Dataset	ETH-80	YTC	Virus	AFEW	MDSD
d_{m1}^{v2}	40	120	30	70	40
d_{m2}^{v2}	9	38	8	15	7
m_1^{v2}	5	3	3	5	4
m_2^{v2}	20	9	9	4	16
η_{r1}^{v2}	5.0	1.0	30.0	1E-6	6.5
η_{r2}^{v2}	5.0	0.1	30.0	1E-6	3.0
ϵ_{r1}^{v2}	3E-3	1E-4	7E-3	1E-3	1E-3
ϵ_{r2}^{v2}	3E-3	1E-3	1E-3	1E-4	1E-2

B. Implementation Details

For the proposed SymNet, we construct its architecture with two different versions. The first version consists of six components: $X_0 \rightarrow f_m \rightarrow f_r \rightarrow f_p \rightarrow f_{\log} \rightarrow$ KDA $\rightarrow$ NN, and we name it SymNet-v1. Its deep version is made up of seven components: $X_0 \rightarrow f_m \rightarrow f_r \rightarrow f_m \rightarrow f_r \rightarrow f_{\log} \rightarrow$ KDA $\rightarrow$ NN, and we call it SymNet-v2. For SymNet-v1, we use d_m^{v1} and m^{v1} to, respectively, represent the dimensionality of the learned new SPD matrices and the number of filters of the f_m layer. The thresholds of the rectifying layer are denoted as η_r^{v1} and ϵ_r^{v1}. Since SymNet-v2 contains two SPD matrix mapping layers and two rectifying layers, we first use d_{m1}^{v2} and d_{m2}^{v2} and m_1^{v2} and m_2^{v2} to denote the dimensionality of the produced new SPD matrices and the number of filters of the two f_m layers, respectively. Then, the thresholds of the two rectifying layers are represented by η_{r1}^{v2}, ϵ_{r1}^{v2}, η_{r2}^{v2}, and ϵ_{r2}^{v2}. The suitable values for these key parameters involved in SymNet-v1 and its deep version SymNet-v2 on the five used data sets are tabulated in Tables II and III, respectively. For training the proposed model, we just use i7-9700 (3.00 GHz) CPU with 16-GB RAM.

In Section III-A, (7) gives the definition of each connection weight W_k^r. Since it is closely related to the function $\text{div}_{d_{k-1},d_k}(V)$, the following steps provide the detailed implementations for this function. First, as mentioned earlier, the set covariance matrix X^i used to represent the ith image set S_i is of size 400×400. Hence, the size of the computed sample covariance matrix $\mathbb{C}$ is also 400×400. Then, the eigenvalue decomposition is applied to $\mathbb{C}$ for the sake of obtaining its corresponding eigenvalue and eigenvector matrices. Afterward, we record the eigenvector matrix in accordance with the descending order of the magnitudes of the eigenvalues. Next, with the given d_k and m_k, the matrix $V(\mathbb{C})$, which is comprised of $d_k \times m_k$ leading eigenvectors, can be extracted. Finally, by successively dividing $V(\mathbb{C})$ into m_k nonoverlapping parts with each part consisting of d_k eigenvectors, the connection weights of SymNet-v1 and Symnet-v2 are defined. Taking SymNet-v1 and ETH-80 data set as an example, due to the values of d_m^{v1} and m^{v1} on this data set which are, respectively, set to 20 and 8, the size and the number of transformation matrices of SymNet-v1 are 400×20 and 8, respectively.

Fig. 3. Face frames of the YTC data set.

Fig. 4. Sample images of the ETH-80 data set.

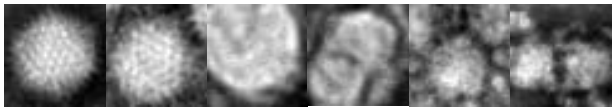


Fig. 5. Sample virus images of the Virus data set.

C. Data Sets Description and Settings

1) YTC Data Set: This data set is made up of 1910 video clips of 47 different subjects that were collected from the website of YouTube. Each clip consists of hundreds of face frames, most of which exhibit a wide range of within-class variations in pose, resolution, illumination, and expression. Some face instances of this data set are shown in Fig. 3.

2) ETH-80 Data Set: This data set is comprised of eight categories, such as cows, cups, horses, dogs, tomatoes, cars, pears, and apples. Each category is made up of ten subcategories, each of which consists of 41 images of different views. Fig. 4 shows some examples of this data set.

3) Virus Data Set: This data set was obtained via transmission electron microscopy (TEM) technique. It is composed of 15 different virus types, each of which contains 100 TEM image patches. Different virus types in this data set have different sizes and shapes. Besides, the virus patches within a given category exhibit some common characteristics, such as constant diameter, low resolution, no sufficient apparent information, and unclear contour information. Some virus images of this data set are presented in Fig. 5. In our experiments, we equally group each virus class into five subclasses for image set classification.

4) MDSD Data Set: This data set is comprised of 13 different categories of dynamic scenes, each of which contains ten different scene sequences collected in unconstrained scenarios. Since the MDSD data set exhibits a wide range of intrasubject variations in physical morphology, background, and illumination, classification over it seems challenging. Fig. 6 shows some scene images of this data set.

5) AFEW Data Set: The acted facial expression in the wild (AFEW) data set involves 1345 video sequences of seven different types of facial expressions, such as angry, disgust, fear, happy, neutral, sad, and surprise. They are collected from movies with close to real-world scenarios. Some instances of this data set are shown in Fig. 7. For a fair comparison, we follow the standard protocols of the Emotion Recognition in the Wild Challenge (EmotiW2014) [46] and [12] to first split these training video sequences into 1746 small clips for data augmentation. Then, the recognition results of the different performers are reported on the validation set as the ground truth of the test set is not publicly available.

Fig. 6. Sample images of the MDSD data set.

Fig. 7. Sample images of the AEFW data set.

TABLE IV

RECOGNITION SCORE (%) COMPARISON ON THE YTC AND AFEW DATA SETS

Methods	YTC	AFEW
AIM [18]	62.85 ± 3.46	21.35
Stein [19]	61.46 ± 3.53	20.81
LEM [17]	63.91 ± 3.25	21.62
GDA [2]	65.78 ± 3.34	29.11
GEDA [48]	66.37 ± 3.52	29.45
CDL [4]	68.76 ± 2.96	31.62
LMKML [24]	70.31 ± 2.66	-
PML [13]	67.62 ± 3.32	28.98
LEML [7]	69.04 ± 3.84	25.13
SPDML-AIM [9]	64.66 ± 2.92	26.72
SPDML-Stein [9]	61.57 ± 3.43	24.55
SPDNet [12]	67.38 ± 2.01	**34.23**
SymNet-v1	73.12 ± 1.55	31.89
SymNet-v2	**73.62 ± 3.16**	32.70

For a fair comparison, we follow the previous works [4]–[8] to prepare our experiments. The first step is to make the training and test samples. Specifically, we randomly select nine video sequences per class of the YTC data set with three for training and six for testing. For the ETH-80 data set, we randomly choose five image sets in each category for gallery and the rest for probes. For the Virus data set, each subject has three randomly selected image sets for training and the rest for the query set. For the MDSD data set, we make use of the seventy-thirty-ratio (STR) protocol, i.e., the gallery and probes are built by randomly choosing seven video clips for the training set and the rest three for the query set in each category. Then, the aforementioned selection process corresponding to each data set is repeated ten times such that ten different pairs of training and test sets can be generated for averaging classification results. Finally, each image of the five used data sets is tailored into a 20×20 grayscale one, and a 400×400 covariance matrix thus can be computed for image set representation.

D. Results and Discussion

According to the experimental results reported in Tables IV and V, we summarize some interesting observations into the following four aspects. First, we want to make a comparison between the three basic Riemannian metrics. It is clear to see that LEM outperforms Stein and AIM in terms of classification result and standard derivation on the YTC and ETH-80 data sets. Besides, the classification accuracies of LEM are also superior to those of Stein and AIM on the AFEW, Virus, and MDSD data sets. These findings demonstrate that LEM is more precise and effective than AIM and Stein in measuring the geodesic distance between any two SPD matrices.

Second, the comparison between CDL, SPDML-AIM/SPDML-Stein, and LEML is what we are also interested in. As can be clearly seen from Tables IV and V, the classification scores of LEML and CDL are significantly higher than that of SPDML-AIM/SPDML-Stein on the YTC, ETH-80, and Virus data sets. This indicates that the LEM-based metric learning approaches exhibit more superiority than AIM- and Stein-based ones in similarity measurement of SPD matrix. Besides, it is also interesting to note that LEML outperforms CDL in terms of classification accuracy on the YTC, ETH-80, and Virus data sets, which experimentally proves that the way of directly performing mapping learning and metric learning on the SPD manifold is able to encode the Riemannian geometry of the original data manifold more faithfully than the Euclidean treatment.

TABLE V

CLASSIFICATION SCORE (%) COMPARISON ON THE ETH-80, VIRUS, AND MDSD DATA SETS

Methods	ETH-80	Virus	MDSD
AIM [18]	87.50 ± 5.77	27.00 ± 4.12	13.08 ± 4.05
Stein [19]	88.00 ± 5.11	25.80 ± 4.68	12.67 ± 4.25
LEM [17]	89.25 ± 4.72	27.67 ± 4.17	13.74 ± 4.52
GDA [2]	93.25 ± 4.80	47.00 ± 2.49	30.51 ± 7.78
GEDA [48]	93.75 ± 3.34	48.67 ± 2.33	30.37 ± 5.16
CDL [4]	93.75 ± 3.43	48.33 ± 3.60	31.28 ± 2.82
LMKML [24]	94.25 ± 3.69	50.19 ± 5.83	32.37 ± 4.53
MMML [5]	95.00 ± 1.89	51.13 ± 7.60	32.56 ± 6.26
PML [13]	93.25 ± 3.54	17.33 ± 4.66	29.67 ± 4.66
LEML [7]	94.00 ± 3.31	55.67 ± 9.94	29.30 ± 3.89
SPDML-AIM [9]	90.75 ± 3.34	40.68 ± 7.00	30.04 ± 5.06
SPDML-Stein [9]	90.50 ± 3.87	42.00 ± 7.24	27.69 ± 4.88
SPDNet [12]	92.50 ± 3.69	65.00 ± 2.36	32.05 ± 1.81
SymNet-v1	96.00 ± 4.54	71.67 ± 7.53	**35.58 ± 8.16**
SymNet-v2	**97.00 ± 2.74**	**73.00 ± 8.67**	34.62 ± 8.70

Third, we would like to make a discussion between LMKML and Multiple Manifolds Metric Learning (MMML). It is intuitive to see that the classification performance of LMKML and MMML outperforms most of the competitors on the YTC, ETH-80, Virus, and MDSD data sets, which suggests that the complementarity of multiple statistics in set data modeling can help to extract more desirable structural information for visual classification. However, MMML exhibits a better classification ability than LMKML on the ETH-80, Virus, and MDSD data sets. The main reason is that LMKML applies a Euclidean kernel function to the non-Euclidean high-order statistics for kernel space embedding, which is unable to preserve the Riemannian properties, and thus may lead to sub-optimal learning results. In contrast, MMML treats different data manifolds with different Riemannian kernel functions.

Finally, the comparison between SPDNet and the proposed SymNet is what we especially care about. As aforementioned, both of them focus on learning a more compact and efficient SPD manifold from the original one. However, SPDNet achieves this goal by exploiting the end-to-end learning mechanism in the context of SPD manifold. For the proposed SymNet, the $(2\text{-D})^2$PCA algorithm is integrated into the designed lightweight cascaded architecture to introduce unsupervised parameter learning. From Table IV, we can find that the recognition ability of the proposed model is superior to all the competitors except for SPDNet on the complicated and relatively large-scale AFEW data set. However, on the remaining four data sets, SPDNet shows a poor classification performance. These experimental observations suggest that SPDNet is not as effective as the proposed model in handling visual classification tasks on the relatively small-scale data sets. Furthermore, it is evident that the learning ability of SymNet-v1 is inferior to that of SymNet-v2 on the YTC,

TABLE VI
ABLATION STUDY FOR EACH DESIGNED COMPONENT OF SYMNET ON THE FIVE USED DATA SETS

Methods	ETH-80	YTC	Virus	AFEW	MDSD
SymNet-1-v1	89.75 ± 4.78	38.40 ± 1.40	42.67 ± 8.13	22.43	18.27 ± 3.74
SymNet-2-v1	96.50 ± 3.79	73.29 ± 3.28	71.11 ± 8.07	32.97	25.96 ± 7.18
SymNet-3-v1	96.50 ± 3.79	74.70 ± 3.27	71.11 ± 7.79	33.51	30.77 ± 8.56
SymNet-4-v1	96.00 ± 4.54	73.12 ± 1.55	71.67 ± 7.53	31.89	35.58 ± 8.16
SymNet-1-v2	87.25 ± 4.63	37.87 ± 2.26	34.67 ± 5.71	19.73	17.52 ± 7.15
SymNet-2-v2	96.50 ± 2.85	72.62 ± 2.41	72.00 ± 10.56	32.16	31.20 ± 4.98
SymNet-3-v2	97.00 ± 2.74	72.48 ± 3.33	72.00 ± 8.34	32.70	31.62 ± 6.00
SymNet-4-v2	97.00 ± 2.74	73.62 ± 3.16	73.00 ± 8.67	32.70	34.62 ± 8.70

AFEW, ETH-80, and Virus data sets, which could indicate that deeper architecture is qualified to make further improvements in filtering out redundant information and learning more effective semantic representations.

E. Computational Complexity Analysis

According to Section III, we can find that the time consumption of SymNet-v1 comes from five aspects: 1) paying $\mathcal{O}(Nd^3 + Nd^2 d_m^{v1} + N(d_m^{v1})^2 d)$ for the unsupervised parameter learning via (2-D)^{2}PCA in the SPD matrix mapping layer; 2) performing nonlinear feature learning in the rectifying layer needs to pay $\mathcal{O}(Nm^{v1}(d_m^{v1})^3)$; 3) conducting pooling operation needs to pay $\mathcal{O}(Nm^{v1}(d_m^{v1})^3)$; 4) paying $\mathcal{O}(Nm^{v1}(d_m^{v1})^3)$ to carry out Log-Euclidean Riemannian computing; and 5) building kernel matrix in the KDA algorithm needs to pay $\mathcal{O}(N(N-1)m^{v1}(d_m^{v1})^2)$. Considering that $m^{v1} \ll N$ and $N \ll d^2$, the computational complexity of SymNet-v1 is $\mathcal{O}(N^2(d_m^{v1})^2 + (d_m^{v1})^2 d + (d_m^{v1})^3 + d^3)$. Similarly, the computational complexity of SymNet-v2 is $\mathcal{O}(d^3 + (d_{m1}^{v2})^3 + N^2(d_{m2}^{v2})^2 + (d_{m2}^{v2})^3)$. For SPDNet, its computation time is mainly consumed in two aspects: 1) building the three BiMap layers and the LogEig layer and 2) performing end-to-end training with the Riemannian matrix backpropagation computing. As a consequence, its computational complexity is $\mathcal{O}(Eh_1^3 + Eh_3^3 + Eh_5^3 + Ed^2 h_1)$. Here, h_i denotes the dimensionality of the generated new features in the ith layer and E is the number of training epochs. Given a typical setting with $d = 400$, $N = 141$, $d_m^{v1} = 60$, $d_{m1}^{v2} = 120$, and $d_{m2}^{v2} = 38$, we can have $N^2(d_m^{v1})^2 + (d_m^{v1})^2 d + (d_m^{v1})^3 + d^3 \approx 10^8$ and $d^3 + (d_{m1}^{v2})^3 + N^2(d_{m2}^{v2})^2 + (d_{m2}^{v2})^3 \approx 10^8$. As for SPDNet, given $d = 400$, $E = 500$, $h_1 = 200$, $h_3 = 100$, and $h_5 = 50$, we can obtain $Eh_1^3 + Eh_3^3 + Eh_5^3 + Ed^2 h_1 \approx 10^{10}$. Therefore, the computational complexity of the proposed SymNet is much lower than that of SPDNet. Note that, the values of these parameters are selected according to the experiments.

F. Visualization

In this section, we choose the Virus data set as an example to perform the 2-D visualization experiments for the sake of verifying the discriminatory power of the data representations learned by SymNet-v1 and SymNet-v2, intuitively. As mentioned earlier, the Virus data set is made up of 15 different virus categories, each of which has three randomly selected image sets for training. The experimental results, obtained by utilizing the t-stochastic neighbor embedding (SNE) technique [49] to embed the original SPD manifold-valued features into a 2-D space, are shown in Fig. 8. Compared with Fig. 8(a), what can be notably found from Fig. 8(c1) is that the samples from the same class have large within-class compactness and the samples from different categories exhibit small between-class similarity. Furthermore, we also visualize the distributions of the lower dimensional SPD matrices, generated by the SPD matrix mapping layer of SymNet-v1 and SymNet-v2, in Fig. 8(b1) and (b2), respectively, and it is evident that most categories can be separated up to a point. This demonstrates that the introduced unsupervised learning scheme for filter banks is feasible and effective. According to Fig. 8(c2), we can also observe that the discriminability of the features learned by SymNet-v2 is superior to that of SymNet-v1 in terms of larger interclass separability on the Virus data set. These observations further confirm the effectiveness of the designed lightweight cascaded network for SPD matrix nonlinear learning.

G. Ablation Study for Each Designed Component of SymNet

To investigate the efficacy of each designed layer in SymNet-v1 and its deep version SymNet-v2, we conduct classification experiments on the five used data sets. The experimental results obtained by different subnetworks of the proposed SymNet-v1 and SymNet-v2 are listed in Table VI. From this table, it is intuitive to see that SymNet-2-v1, which makes up of the SPD matrix mapping layer and log-map layer, yields a significant improvement in classification score on all the used data sets compared with SymNet-1-v1 that just contains the SPD matrix mapping layer. This demonstrates the importance of Riemannian computing in preserving the geometrical structure of the raw data in new feature space. Based on SymNet-2-v1, the rectifying layer is added between the SPD matrix mapping layer and the log-map layer. As a consequence of this measure, the classification performance of SymNet-3-v1 has been further promoted on the YTC, Virus, AFEW, and MDSD data sets, which justifies its effectiveness in enhancing the discriminatory power of the learned representations. According to Table VI, we can also note that SymNet-4-v1, which is constituted by coupling the suggested SPD matrix pooling layer with SymNet-3-v1, achieves competitive classification performance on the five used data sets. This experimentally certifies its feasibility in compressing the geometric features.

From the last four lines of Table VI, we can note that after integrating the log-map layer onto the tail of the second SPD matrix mapping layer of SymNet-1-v2, the classification ability of SymNet-2-v2 has been greatly lifted on all the used data sets. This again validates the significance of Riemannian computing. Another interesting observation is that the classification performance of SymNet-3-v2, which is constructed by appending a rectifying layer between the two SPD matrix mapping layers of SymNet-2-v2, is superior to that of SymNet-2-v2 on the ETH-80, AFEW, and MDSD data sets. Based on SymNet-3-v2, another rectifying layer is added between the second SPD matrix mapping layer and the log-map layer. As a result of this manipulation, SymNet-4-v2 outperforms SymNet-3-v2 in classification result on the YTC, Virus, and MDSD data sets. Meanwhile, its classification performance is the same as that of SymNet-3-v2 on the remaining two data

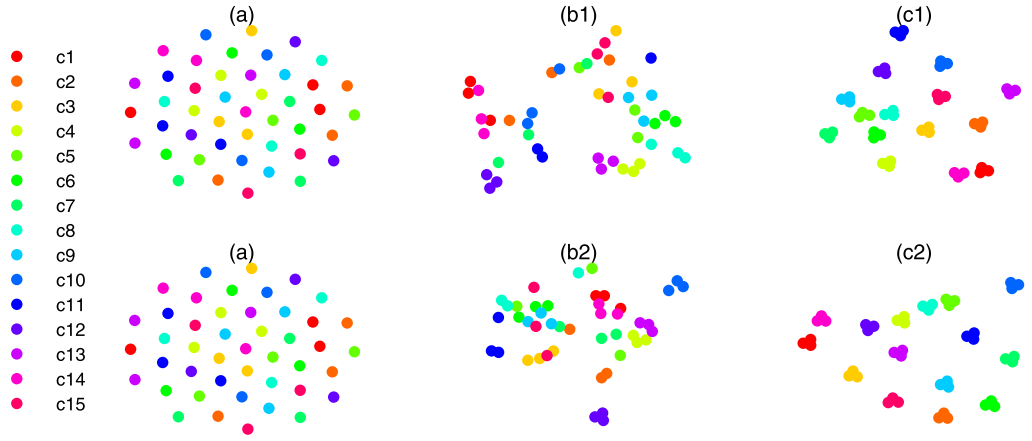


Fig. 8. 2-D visualization of the learned feature representations of SymNet-v1 and SymNet-v2 on the Virus data set, where colors indicate categories. (a) Original SPD manifold-valued data distribution. (b1) and (b2) Distributions of the generated lower dimensional SPD matrices of SymNet-v1 and SymNet-v2. (c1) and (c2) Sample distributions after using SymNet-v1 and SymNet-v2.

TABLE VII

ABLATION STUDY FOR THE POOLING STRATEGY ON THE FIVE USED DATA SETS

Methods	ETH-80	YTC	Virus	AFEW	MDSD
SymNet-TMaP-v1	96.00 ± 4.54	67.59 ± 2.66	71.67 ± 7.53	18.11	35.58 ± 8.16
SymNet-TMeP-v1	95.00 ± 3.95	73.12 ± 1.55	65.00 ± 4.59	31.89	26.92 ± 7.87
SymNet-TMaP-v2	89.75 ± 5.06	54.82 ± 1.89	61.00 ± 11.55	28.38	26.74 ± 2.50
SymNet-TMeP-v2	89.25 ± 3.34	71.91 ± 2.44	59.67 ± 9.62	32.16	23.81 ± 4.85
SymNet-NP-v2	97.00 ± 2.74	73.62 ± 3.16	73.00 ± 8.67	32.70	34.62 ± 8.70

sets because of the output symmetric matrices of its second SPD matrix mapping layer which are highly nonsingular. These findings further prove the validity of the designed rectifying layer in SPD matrix nonlinear learning.

H. Ablation Study for the Pooling Strategy

As is well known, both the conventional max- and mean-pooling operations are widely used in the field of ConvNets. In order to evaluate which one is more suitable for the proposed tangent space pooling tactic, we make experiments on the five used data sets to observe the classification performance of SymNet-v1 and SymNet-v2 versus different pooling tactics. The experimental results are listed in Table VII, where "TMaP" and "TMeP" represent the tangent space max- and mean-pooling operations, respectively. From this table, we can see that the classification performance of SymNet-TMaP-v1 is superior to that of SymNet-TMeP-v1 on the ETH-80, Virus, and MDSD data sets, while it is reversed on the YTC and AFEW data sets. The same observation can also be found between SymNet-TMaP-v2 and SymNet-TMeP-v2. Despite this, the classification ability of SymNet-TMaP-v2 and SymNet-TMeP-v2 is surpassed by SymNet-NP-v2 on all the used data sets. Here, "NP" means no pooling operation. This may indicate that the proposed tangent space pooling method is inappropriate for deeper Riemannian network because of the distortion of the local geometrical structure of the features in the mapping process. We need to emphasize that the pooling layer of SymNet-TMaP-v2 and SymNet-TMeP-v2 is added between the first rectifying layer and the second SPD matrix mapping layer of SymNet-v2, mainly because the size of the resulting SPD matrices of its second rectifying layer is too small to perform pooling operation. Currently, this is also considered to be the main bottleneck to keep the proposed SymNet from going deeper.

In addition to the aforementioned experimental analyses, in this section, we take the ETH-80, MDSD, and Virus data sets as three examples to further investigate the impact of another two pooling strategies, studied in Section III-C, on the classification performance of our approach. The experimental results of SymNet-SMAP-v1, SymNet-SMAP-v2, SymNet-SMeP-v1, and SymNet-SMeP-v2 on these three data sets are tabulated in Table VIII, where "SMaP" and "SMeP," respectively, denote the max- and mean-pooling operations that are directly imposed on the SPD manifold. As can be apparently seen from Table VIII, the classification scores of SymNet-SMAP-v1 are significantly lower than those of SymNet-SMeP-v1 on the ETH-80, Virus, and MDSD data sets. The reason may be that the sliding window of max-pooling operation does not move along the geodesic, thus destroying the Riemannian geometry of the SPD manifold to a certain extent. From Table VIII, it is also worth noting that although SymNet-SMeP-v2 outperforms SymNet-SMAP-v2 in classification performance on the Virus and the MDSD data sets, its classification ability is distinctly inferior to that of SymNet-NP-v2 of Table VII. This again suggests that pooling operation is currently inapplicable to SymNet-v2.

TABLE VIII

AVERAGE CLASSIFICATION SCORES (%) OF THE PROPOSED MODEL UNDER DIFFERENT POOLING STRATEGIES ON THE ETH-80, MDSD, AND VIRUS DATA SETS

Methods	ETH-80	Virus	MDSD
SymNet-SMaP-v1	70.50 ± 7.38	52.78 ± 7.43	11.54 ± 4.94
SymNet-SMeP-v1	96.00 ± 2.85	72.78 ± 7.43	29.49 ± 4.75
SymNet-SMaP-v2	87.75 ± 8.45	60.00 ± 7.37	19.41 ± 4.40
SymNet-SMeP-v2	86.75 ± 3.55	64.33 ± 8.61	26.74 ± 3.26

For the proposed model, despite that "SMeP" is a competitive pooling method, it is more time-consuming than the tangent space pooling tactic (i.e., "TMaP" and "TMeP") because of optimization. In consequence, the "TMeP" is adopted to the proposed SymNet-v1 on the YTC and AFEW data sets. On the remaining three data sets, the "TMaP" seems

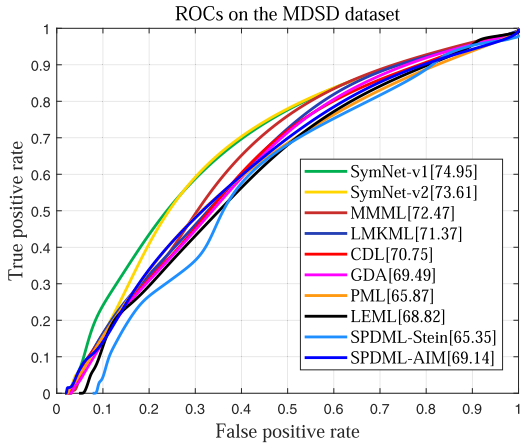


Fig. 9. ROC curves of the different methods on the MDSD data set.

to be a better choice for SymNet-v1. In contrast, we do not embed any pooling operations into SymNet-v2. However, the pooling strategies mentioned earlier are just an attempt and exploration on how to generalize the paradigm of conventional pooling operations to the SPD manifold for DR and feature selection. More investigations and works should be done in the future.

I. Ablation Study for Different Validation Metrics

According to the previous studies, we can see that both SymNet-v1 and its deep version SymNet-v2 show their effectiveness in video-based image set classification. However, the overall accuracy and the standard derivation are the only two validation metrics used in this article. To further assess our model, we conduct experiments on the MDSD data set to compare our approach with several representative image set classification methods under other commonly used validation metrics [50], [51], such as micro-based average precision ($\text{Precision}_{\text{mi}}$), macro-based average precision ($\text{Precision}_{\text{ma}}$), micro-based average recall ($\text{Recall}_{\text{mi}}$), macro-based average recall ($\text{Recall}_{\text{ma}}$), micro-based average specificity ($\text{Specificity}_{\text{mi}}$), macro-based average specificity ($\text{Specificity}_{\text{ma}}$), micro-based F1-Score ($\text{F1-Score}_{\text{mi}}$), and macro-based F1-Score ($\text{F1-Score}_{\text{ma}}$). The classification scores of the different methods on this data set are tabulated in Table IX. From Table IX, we can note that SymNet-v1 and SymNet-v2 are the best two performers on the MDSD data set.

In addition, we also draw the ROC curves for the proposed model and the competitors on this data set to intuitively test their reliability in video-based image set classification. The experimental results are shown in Fig. 9. From this figure, it is evident to see that the AUCs of SymNet-v1 and SymNet-v2 are, respectively, 74.95 and 73.61, larger than that of all the comparative methods. This again shows that the proposed lightweight cascaded network for SPD matrix nonlinear leaning is valid.

J. Ablation Study for the Designed Nonlinear Activation Function f_{r_l} in (9)

To verify the effectiveness of the designed nonlinear activation function f_{r_1}, we take the MDSD data set as an example to study the impact of the activation threshold η on the classification performance of SymNet-v1 and its deep version SymNet-v2, respectively. The classification score of SymNet-v1 versus different values of η_r^{v1} on this data set

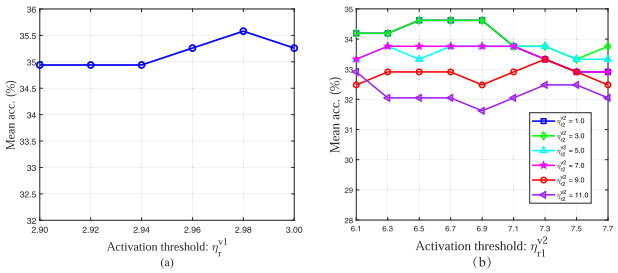


Fig. 10. Average classification results of (a) SymNet-v1 and (b) SymNet-v2 on the MDSD data set under different parameter settings.

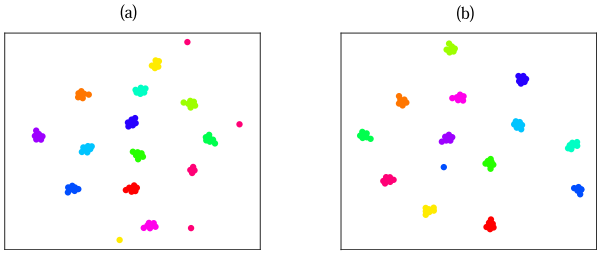


Fig. 11. 2-D visualization of the features learned by SymNet-v1. (a) $\eta_r^{v1} = 0.00$. (b) $\eta_r^{v1} = 2.98$.

is shown in Fig. 10(a). From this figure, it is evident that the proposed SymNet-v1 obtains the best classification result on the MDSD data set when η_r^{v1} is set to 2.98. Fig. 10(b) shows the classification score of SymNet-v2 versus different η_{r1}^{v2} and η_{r2}^{v2} on the MDSD data set. From Fig. 10(b), we can observe that when η_{r1}^{v2} and η_{r2}^{v2}, respectively, vary in the range of {6.5,6.7,6.9} and {1.0,3.0}, the proposed SymNet-v2 exhibits the best classification performance. According to the results presented in Fig. 10(a) and (b), it is also worth noting that SymNet-v1 and SymNet-v2 tend to be less sensitive to the activation threshold η. Here, we want to state that when the proposed network does not include this activation function, the classification results of SymNet-v1 and SymNet-v2 obtained on this data set are 35.26% and 34.19%, lower than that of the best cases discussed above. These experimental observations suggest that f_{r_1} could play a part in ameliorating the discriminability of the learned feature representations. For SymNet-v1 and SymNet-v2, η_r^{v1}, η_{r1}^{v2}, and η_{r2}^{v2} are configured as 2.98, 6.5, and 3.0 on the MDSD data set, respectively. Their corresponding values on the remaining four data sets are tabulated in Tables II and III.

In order to evaluate the validity of f_{r_1} more intuitively, we further conduct the 2-D visualization experiments on the MDSD data set to study the influence of using and without using this activation function on the data distribution learned by SymNet-v1 and SymNet-v2. The experimental results, obtained via the t-SNE technique, are drawn in Figs. 11 and 12. Here, different colors and points indicate different categories and image set samples, respectively. As can be clearly seen, after integrating f_{r_1} into the rectifying layers of the proposed model, the problem of within-class diversity reflected in Figs. 11(a) and 12(a) has been further relieved, respectively, incarnated in Figs. 11(b) and 12(b). The aforementioned experimental observations demonstrate the significance of f_{r_1} in improving the representation ability of our model to a certain extent.

K. Parameter Discussion

As stated in Section III, the number of filters of the SPD matrix mapping layer and the thresholds of the rectifying layer

TABLE IX

AVERAGE CLASSIFICATION RESULTS (%) OF THE PROPOSED MODEL AND SOME REPRESENTATIVE IMAGE SET CLASSIFICATION METHODS ON THE MDSD DATA SET UNDER DIFFERENT VALIDATION METRICS

Methods	Overall accuracy	Precision_{mi}	Recall_{mi}	Specificity_{mi}	Precision_{ma}	Recall_{ma}	Specificity_{ma}	F1-Score_{mi}	F1-Score_{ma}
CDL [4]	31.28	31.28	31.28	94.27	29.51	31.28	94.27	31.28	30.27
GDA [2]	30.51	30.51	30.51	94.21	31.18	30.51	94.21	30.51	30.79
PML [13]	29.67	29.67	29.67	94.14	30.33	29.67	94.14	29.67	27.58
LEML [7]	29.30	29.30	29.30	94.11	27.71	29.30	94.11	29.30	27.29
SPDML-AIM [9]	30.04	30.04	30.04	94.17	30.52	30.04	94.17	30.04	30.96
SPDML-Stein [9]	27.69	27.69	27.69	93.97	28.31	27.69	93.97	27.69	30.02
LMKML [24]	32.37	32.37	32.37	94.36	31.29	32.37	94.36	32.37	31.67
MMML [5]	32.56	32.56	32.56	94.38	33.71	32.93	94.38	32.56	32.93
SymNet-v1	**33.58**	**35.58**	**35.58**	**94.63**	**36.56**	**35.58**	**94.63**	**35.58**	**35.61**
SymNet-v2	34.62	34.62	34.62	94.55	34.07	34.62	94.55	34.62	34.25

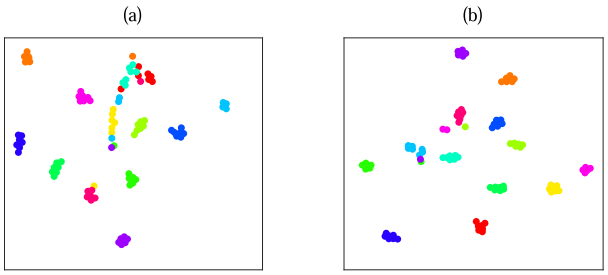


Fig. 12. 2-D visualization of the features learned by SymNet-v2. (a) $\eta_{r1}^{v2} = 0.00$; $\eta_{r2}^{v2} = 0.00$. (b) $\eta_{r1}^{v2} = 6.50$; $\eta_{r2}^{v2} = 3.00$.

TABLE X

AVERAGE CLUSTERING RESULTS OF THE DIFFERENT METHODS ON THE ETH-80 DATA SET UNDER DIFFERENT VALIDATION METRICS, WHERE "CA" REPRESENTS THE CLUSTERING ACCURACY

Methods	CA	Precision_{mi}	Recall_{mi}	F1-Score_{mi}
SSC	91.00	82.29	84.75	83.49
SymNet-v1-SSC	92.50	83.12	85.94	84.50
SymNet-v2-SSC	93.75	86.16	86.88	86.83

are some of the key factors of the proposed SymNet. In what follows, we take the MDSD data set as an example to carry out experiments to study the impact of them on the classification performance of SymNet-v1 (due to space limitation, we do not investigate them in the context of SymNet-v2). According to the classification results presented in Fig. 13, we can see that the values of m^{v1} and ϵ_r^{v1} should be configured as 4 and 1E−3 on this data set, respectively. As can be noted from Fig. 13(a), when the value of m^{v1} is surpassed 4, the classification ability of SymNet-v1 tends to be degraded because of the gradually increased redundant information of the learned features. From Fig. 13(b), we can also observe that the smaller the value of ϵ_r^{v1}, the worse the exhibited classification performance of SymNet-v1. This experimentally certifies the availability of f_{r_2} in introducing nonlinearity for the mappings. Since the value of d_m^{v1} is determined by the (2-D)^{2}PCA algorithm, we do not discuss it here. Please kindly refer to Tables II and III for their corresponding values on other data sets.

L. Unsupervised SPD Matrix Learning

As introduced in Section III-A, the proposed model makes use of (2-D)^{2}PCA to perform unsupervised filter learning. Accordingly, in this section, we take the ETH-80 data set as an example to further study its availability in the scenario of without accessing to the labels of the training samples. To tackle this unsupervised SPD matrix learning problem, the following setups are made for the proposed approach. First, the KDA algorithm is removed from the proposed model.

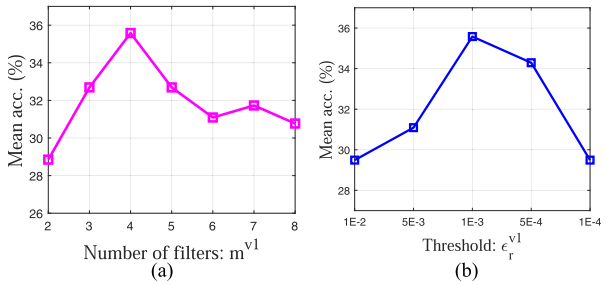


Fig. 13. Classification results of SymNet-v1 on the MDSD data set under different parameter settings. (a) Classification score of SymNet-v1 versus m^{v1}. (b) Classification score of SymNet-v1 versus ϵ_r^{v1}.

Second, the unlabeled training SPD matrices are used to train SymNet-v1 and SymNet-v2, respectively. After the training process is completed, the lower dimensional data representations can be extracted for the test samples via the trained networks. Finally, the classical sparse subspace clustering (SSC) [52] algorithm is exploited to test the learning ability of our approach in this context. The clustering results of SSC, SymNet-v1-SSC, and SymNet-v2-SSC on this data set are shown in Table X. From this table, it is obvious that the clustering performance of SymNet-v1-SSC and SymNet-v2-SSC is superior to that of SSC under all the validation metrics on this data set. This again demonstrates the effectiveness of the proposed approach in SPD matrix learning. To prepare data for SSC, we first simplify the proposed model to just contain an SPD matrix mapping layer, used to conduct DR with 99% energy preservation of the training data, and a log-map layer for Riemannian computation. Besides, the number of feature maps of these two layers is set to 1. With this very simplified architecture, the data applicable to SSC can be obtained.

M. Computational Time Comparison

To show the time efficiency of the proposed SymNet, we finally make experiments on the YTC data set to compare its average training and testing time with some representative image set classification methods. The experiments were run on 3.0 GHz PC with 16-GB RAM, and the computation time of each method obtained with the MATLAB2019a software is tabulated in Table XI. We need to emphasize that the running time is reflected by the CPU time, and the testing time is computed by classifying one query set with all the training samples. According to Table XI, we have three interesting observations listed as follows. First, the training time of the proposed SymNet-v2 is higher than that of CDL and GDA. The main reason is that SymNet-v2 needs to pay more time to conduct the two-staged unsupervised parameter learning. This reason can also be taken to explain the difference

TABLE XI

AVERAGE RUNNING TIME (SECONDS) OF THE DIFFERENT METHODS ON THE YTC DATA SET (CLASSIFICATION OF ONE VIDEO)

Methods	Training	Testing
CDL [4]	29.79	0.23
GDA [2]	57.31	0.39
GEDA [48]	76.80	0.59
PML [13]	660.89	0.05
LEML [7]	192.57	0.61
SPDML-AIM [9]	871.62	1.25
SPDML-Stein [9]	332.73	0.62
SPDNet [12]	1668.06	0.06
SymNet-v1	23.36	0.05
SymNet-v2	70.03	0.08

between SymNet-v1 and SymNet-v2 in running time. Besides, the dimensionality of the features produced by the first SPD matrix mapping layer of SymNet-v2 is higher than that of SymNet-v1 on the YTC data sets, which can be treated as another reason. Second, what can be clearly found from Table XI is the training burden of SymNet-v1 and SymNet-v2 has been significantly reduced compared to SPDNet, SPDML, LEML, and PML. This again proves the utility of exploiting $(2\text{-D})^2$PCA to train the designed lightweight cascaded SPD matrix learning network. Finally, the testing time of SymNet-v1 and SymNet-v2 is lower than almost all the competitors, which further justifies the practicability of the proposed approach.

V. APPLICATION TO 3-D HAND ACTION RECOGNITION

Recently, 3-D hand action recognition has made more modest progress in first-person view due to its huge possibilities for practical application and the recent availability of RGB-D sensors. It concerns the task of automatically comprehending an action sequence via the 3-D coordinates of hand joints to recognize what first-person hand action interacting with 3-D object is being performed. As each video sequence can be viewed as an image set, this problem can also be solved from the perspective of image set classification. In this section, we further evaluate the validity of the proposed approach on this task using the FPHA data set [53].

The FPHA is a large and diverse first-person hand action data set for 3-D hand pose estimation. It is comprised of 1175 hand action videos belonging to 45 different categories, acted by six actors in three different scenarios. Some hand gesture instances of this data set are presented in Fig. 14. Due to a wide range of intrasubject variability of scale, speed, style, and viewpoint covered in the hand action video sequences, recognition on this data set seems challenging. For evaluations, we follow the standard protocol of [53] to first normalize each video clip to contain 50 frames. Then, each hand gesture frame is characterized by a 63-D feature vector converted from the 3-D coordinates of 21 hand joints provided. As a consequence, a covariance matrix of the size 63×63 can be computed to represent each video sequence. Finally, we follow the 1:1 setting, i.e., 600 action sequences for training and the remaining 575 for testing, to carry out experiments.

To better validate our model, we select some state-of-the-art hand action recognition methods for comparison, such as convolutional two-stream network (two stream) [54], novel view [55], LSTM [53], hierarchical recurrent neural network (HBRNN) [57], jointly learning heterogeneous features (JOULE) [58], transition forests (TF) [59], temporal convolution network (TCN) [60], and unified hand and object model

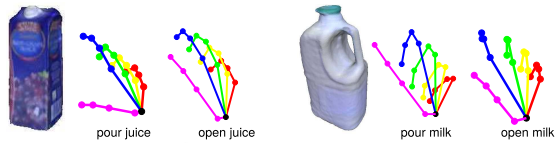


Fig. 14. Some hand action instances of the FPHA data set.

TABLE XII

RECOGNITION SCORE (%) COMPARISON ON THE FPHA DATA SET

Method	Year	Color	Depth	Pose	Accuracy
Two stream-color [54]	2016	✓	✗	✗	61.56
Two stream-flow [54]	2016	✓	✗	✗	69.91
Two stream-all [54]	2016	✓	✗	✗	75.30
Novel View [55]	2016	✗	✓	✗	69.21
1-layer LSTM [53]	2018	✗	✗	✓	78.73
2-layer LSTM [53]	2018	✗	✗	✓	80.14
HBRNN [57]	2015	✗	✗	✓	77.40
JOULE-color [58]	2015	✓	✗	✗	66.78
JOULE-pose [58]	2015	✗	✗	✓	74.60
JOULE-all [58]	2015	✓	✓	✓	78.78
TF [59]	2017	✗	✗	✓	80.69
TCN [60]	2017	✗	✗	✓	78.57
H+O [61]	2019	✓	✗	✗	82.43
LEML [7]	2015	✗	✗	✓	79.48
SPDML-AIM [9]	2018	✗	✗	✓	78.40
SPDNet [12]	2017	✗	✗	✓	**83.79**
SymNet-v1		✗	✗	✓	81.04
SymNet-v2		✗	✗	✓	**82.96**

(H+O) [61]. Besides, we also compare it with three representative SPD manifold learning algorithms, including LEML [7], SPDML-AIM [9], and SPDNet [12]. The recognition scores of the different methods are tabulated in Table XII. Note that, we run the methods of LEML, SPDML-AIM, SPDNet, and TCN with their publicly available source codes and report their best experimental results on this data set. As for H+O and other performers, their recognition scores are from [61] and [53], respectively.

From Table XII, it is clear to see that LEML and SPDML-AIM show comparable recognition performance with other competitors on the FPHA data set. This again demonstrates the effectiveness of Riemannian geometry in characterizing the nonlinear structure of the visual data. From this table, we can also find that SPDNet obtains the highest recognition score on this data set, which provides further confirmation of the significance of nonlinear deep mapping learning for SPD matrix in mining fine-grained geometric representations for visual scenarios. Although the recognition ability of the proposed SymNet-v1 and its deep version SymNet-v2 is inferior to that of SPDNet, they still exhibit the competitive classification performance on this data set, with the merit of much lower computational burden. This can also prove the availability of the suggested SPD matrix learning approach.

VI. CONCLUSION

In this article, we present a lightweight cascaded SPD manifold network (SymNet) for developing a possibility of SPD matrix learning. For the proposed SymNet, we discuss how to build the SPD matrix mapping layer, rectifying layer, SPD matrix pooling layer, and log-map layer for the purpose of extracting more discriminative feature representations while preserving the Riemannian geometry of the data manifold simultaneously. Like the conventional deep learning models, the proposed SymNet needs to learn the optimal values for the key parameters, such as the target dimensionality of the resulting SPD matrices, the number of filters, and the activation thresholds. Once these parameters are fixed, training

this lightweight cascaded SPD manifold network with the $(2\text{-D})^2$PCA algorithm becomes very easy and efficient.

Compared with the end-to-end SPD matrix learning network SPDNet, both the proposed SymNet-v1 and its deep version SymNet-v2 surpass it in terms of classification accuracy on the ETH-80, YTC, Virus, and MDSD data sets. This means that our SymNet is able to eliminate the intraclass variations to some extent, thus giving competitive classification performance. However, on the relatively large-scale and challenging AFEW data set, SymNet-v1 and SymNet-v2 do not seem to be sufficient to address the data variability because of the unsupervised filter learning mode and the lightweight structure. Accordingly, integrating the label information into the filter learning process or constructing a deeper SymNet are considered to be two feasible ways to accommodate the aforementioned issues in the future.

In addition, we can also note that the proposed approach failed to make a substantial breakthrough in the classification performance on the complicated AFEW and MDSD data sets. The reason may be that the input SPD matrices of the proposed model are directly computed from the original image sets without containing any representation learning for the images themselves. Consequently, some stubborn redundant information and implicit data variability information cannot be effectively eliminated in the process of SPD matrix learning. As a prospective countermeasure, some preprocessings might be needed before modeling, such as learning deep features for the original images or jointly performing image feature learning and SPD matrix learning within a designed network.

The evaluations on six typical video-based image set classification tasks confirm the feasibility and effectiveness of our approach for SPD matrix nonlinear learning. Furthermore, the proposed SymNet can be served as a valuable baseline for appraising the advanced learning algorithms in the domain of image set classification, especially for challenging visual classification tasks.

REFERENCES

[1] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, "Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2048–2057.

[2] J. Hamm and D. D. Lee, "Grassmann discriminant analysis: A unifying view on subspace-based learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 376–383.

[3] Z. Wu, Y. Huang, and L. Wang, "Learning representative deep features for image set analysis," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1960–1968, Nov. 2015.

[4] R. Wang, H. Guo, L. S. Davis, and Q. Dai, "Covariance discriminative learning: A natural and efficient approach to image set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2496–2503.

[5] R. Wang, X.-J. Wu, K.-X. Chen, and J. Kittler, "Multiple manifolds metric learning with application to image set classification," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 627–632.

[6] Z. Gao, Y. Wu, M. Harandi, and Y. Jia, "A robust distance measure for similarity-based classification on the SPD manifold," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3230–3244, Sep. 2020.

[7] Z. Huang, R. Wang, S. Shan, X. Li, and X. Chen, "Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 720–729.

[8] G. Cheng, P. Zhou, and J. Han, "Duplex metric learning for image set classification," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 281–292, Jan. 2018.

[9] M. Harandi, M. Salzmann, and R. Hartley, "Dimensionality reduction on SPD manifolds: The emergence of geometry-aware methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 48–62, Jan. 2018.

[10] R. Wang, X.-J. Wu, and J. Kittler, "Graph embedding multi-kernel metric learning for image set classification with Grassmann manifold-valued features," *IEEE Trans. Multimedia*, early access, Mar. 18, 2020, doi: 10.1109/tmm.2020.2981189.

[11] M. Faraki, M. T. Harandi, and F. Porikli, "A comprehensive look at coding techniques on Riemannian manifolds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5701–5712, Nov. 2018.

[12] Z. Huang and L. V. Gool, "A Riemannian network for SPD matrix learning," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2017, pp. 2036–2042.

[13] Z. Huang, R. Wang, S. Shan, and X. Chen, "Projection metric learning on Grassmann manifold with application to video based face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 140–149.

[14] H. Sun, X. Zhen, Y. Zheng, G. Yang, Y. Yin, and S. Li, "Learning deep match kernels for image-set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3307–3316.

[15] X. S. Nguyen, L. Brun, O. Lezoray, and S. Bougleux, "A neural network based on SPD manifold learning for skeleton-based hand gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12036–12045.

[16] W. Wang, R. Wang, S. Shan, and X. Chen, "Discriminative covariance oriented representation learning for face recognition with image sets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5599–5608.

[17] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 1, pp. 328–347, Feb. 2007.

[18] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *Int. J. Comput. Vis.*, vol. 66, no. 1, pp. 41–66, Jan. 2006.

[19] S. Sra, "Positive definite matrices and the S-divergence," *Proc. Amer. Math. Soc.*, vol. 144, no. 7, pp. 2787–2797, Jul. 2016.

[20] M. T. Harandi, R. Hartley, B. Lovell, and C. Sanderson, "Sparse coding on symmetric positive definite manifolds using bregman divergences," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1294–1306, Jun. 2016.

[21] R. Vemulapalli and D. W. Jacobs, "Riemannian metric learning for symmetric positive definite matrices," 2015, *arXiv:1501.02393*. [Online]. Available: http://arxiv.org/abs/1501.02393

[22] L. Zhou, L. Wang, J. Zhang, Y. Shi, and Y. Gao, "Revisiting metric learning for SPD matrix based visual representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3241–3249.

[23] M. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell, "Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach," in *Proc. Euro. Conf. Comput. Vis. (ECCV)*, 2012, pp. 216–229.

[24] J. Lu, G. Wang, and P. Moulin, "Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 329–336.

[25] R. Wang, X.-J. Wu, K.-X. Chen, and J. Kittler, "Multiple Riemannian manifold-valued descriptors based image set classification with multi-kernel metric learning," *IEEE Trans. Big Data*, early access, Mar. 20, 2020, doi: 10.1109/tbdata.2020.2982146.

[26] Z. Huang, R. Wang, S. Shan, and X. Chen, "Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning," *Pattern Recognit.*, vol. 48, no. 10, pp. 3113–3124, Oct. 2015.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2012, pp. 1097–1105.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

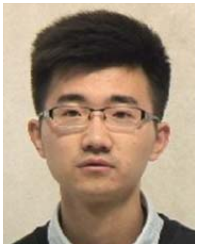
[29] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[30] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, "Background prior-based salient object detection via deep reconstruction residual," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1309–1321, Aug. 2015.

[31] J. Han, X. Yao, G. Cheng, X. Feng, and D. Xu, "P-CNN: Part-based convolutional neural networks for fine-grained visual categorization," *IEEE Trans. Pattern Anal. Mach. Intell.*, early accessAug. 6, 2020, doi: 10.1109/tpami.2019.2933510.

[32] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on the Riemannian manifold of symmetric positive definite matrices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 73–80.

[33] R. Vemulapalli, J. K. Pillai, and R. Chellappa, "Kernel learning for extrinsic classification of manifold features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1782–1789.
[34] J. Zhang, L. Wang, L. Zhou, and W. Li, "Learning discriminative stein kernel for SPD matrices and its applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 1020–1033, May 2016.
[35] M. Harandi and M. Salzmann, "Riemannian coding and dictionary learning: Kernels to the rescue," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3926–3935.
[36] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, "Multi-manifold deep metric learning for image set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1137–1145.
[37] C. Ionescu, O. Vantzos, and C. Sminchisescu, "Training deep networks with structured layers by matrix backpropagation," 2015, *arXiv:1509.07838*. [Online]. Available: http://arxiv.org/abs/1509.07838
[38] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 37–45.
[39] Z. Huang, J. Wu, and L. V. Gool, "Building deep networks on grassmann manifolds," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 3279–3286.
[40] X. Chen, J. Weng, W. Lu, J. Xu, and J. Weng, "Deep manifold learning combined with convolutional neural networks for action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 3938–3952, Sep. 2018.
[41] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, pp. 303–353, Jan. 1998.
[42] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, no. 10, pp. 2385–2404, Oct. 2000.
[43] D. Zhang and Z.-H. Zhou, "$(2D)^2$PCA: Two-directional two-dimensional PCA for efficient face representation and recognition," *Neurocomputing*, vol. 69, nos. 1–3, pp. 224–231, Dec. 2005.
[44] G. Kylberg, M. Uppström, and I. M. Sintorn, "Virus texture analysis using local binary patterns and radial density profiles," in *Proc. Iberoamerican Congr. Pattern Recognit.*, 2011, pp. 573–580.
[45] N. Shroff, P. Turaga, and R. Chellappa, "Moving vistas: Exploiting motion for describing scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1911–1918.
[46] A. Dhall, R. Goecke, J. Joshi, K. Sikka, and T. Gedeon, "Emotion recognition in the wild challenge 2014: Baseline, data and protocol," in *Proc. 16th Int. Conf. Multimodal Interact. (ICMI)*, 2014, pp. 461–466.
[47] T.-K. Kim, J. Kittler, and R. Cipolla, "Discriminative learning and recognition of image set classes using canonical correlations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1005–1018, Jun. 2007.
[48] M. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell, "Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 2705–2712.
[49] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[50] A. Shrivastava, A. K. Tripathy, and P. K. Dalal, "A SVM-based classification approach for obsessive compulsive disorder by oxidative stress biomarkers," *J. Comput. Sci.*, vol. 36, Sep. 2019, Art. no. 101023.
[51] W. D. Fisher, T. K. Camp, and V. V. Krzhizhanovskaya, "Anomaly detection in earth dam and levee passive seismic data using support vector machines and automatic feature selection," *J. Comput. Sci.*, vol. 20, pp. 143–153, May 2017.
[52] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
[53] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with RGB-D videos and 3D hand pose annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 409–419.
[54] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941.
[55] H. Rahmani and A. Mian, "3D action recognition from novel viewpoints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1506–1515.
[56] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 716–723.
[57] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1110–1118.
[58] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5344–5352.
[59] G. Garcia-Hernando and T.-K. Kim, "Transition forests: Learning discriminative temporal transitions for action recognition and detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 432–440.
[60] T. S. Kim and A. Reiter, "Interpretable 3D human action analysis with temporal convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1623–1631.
[61] B. Tekin, F. Bogo, and M. Pollefeys, "H+O: Unified egocentric recognition of 3D hand-object poses and interactions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4511–4520.



Rui Wang received the M.S. degree from the School of Internet of Things Engineering, Jiangnan University, Wuxi, China, in 2018, where he is currently pursuing the Ph.D. degree with the Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence.

His research topics include Riemannian manifold learning, metric learning, and deep learning.

Xiao-Jun Wu received the B.Sc. degree in mathematics from Nanjing Normal University, Nanjing, China, in 1991, and the M.S. and Ph.D. degrees in pattern recognition and intelligent system from the Nanjing University of Science and Technology, Nanjing, in 1996 and 2002, respectively.

From 1996 to 2006, he taught at the School of Electronics and Information, Jiangsu University of Science and Technology, where he was promoted to Professor. He has been with the School of Information Engineering, Jiangnan University since 2006, where he is a Professor of pattern recognition and computational intelligence. He was a Visiting Researcher with the Centre for Vision, Speech, and Signal Processing (CVSSP), University of Surrey, Guildford, U.K., from 2003 to 2004. He has published over 300 papers in his fields of research. His current research interests include pattern recognition, computer vision, and computational intelligence.

Dr. Wu was a Fellow of the International Institute for Software Technology, United Nations University, from 1999 to 2000. He was a recipient of the Most Outstanding Postgraduate Award from the Nanjing University of Science and Technology.

Josef Kittler (Life Member, IEEE) received the B.A., Ph.D., and D.Sc. degrees from the University of Cambridge, Cambridge, U.K., in 1971, 1974, and 1991, respectively.

He is a Distinguished Professor of Machine Intelligence with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford, U.K. He conducts research in biometrics, video and image database retrieval, medical image analysis, and cognitive vision. He published the textbook *Pattern Recognition: A Statistical Approach* and over 700 scientific papers. His publications have been cited more than 70 000 times (Google Scholar).

Dr. Kittler is a Series Editor of Springer Lecture Notes on Computer Science. He currently serves on the Editorial Boards of *Pattern Recognition Letters*, *Pattern Recognition and Artificial Intelligence*, *Pattern Analysis and Applications*. He also served as a member of the Editorial Board of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE during 1982–1985. He served on the Governing Board of the International Association for Pattern Recognition (IAPR) as one of the two British representatives during the period 1982–2005, President of the IAPR during 1994–1996.