Graph Embedding Multi-Kernel Metric Learning for Image Set Classification With Grassmannian Manifold-Valued Features

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

***Abstract*—In the domain of video-based image set classification, a considerable advance has been made by modeling a sequence of video frames (image set) as a linear subspace, which typically resides on a Grassmannian manifold. As a consequence of the large intra-class variations of the video data, there are two open challenges for the modeling task: how to establish appropriate image set models to encode these variations, and how to effectively measure the similarity between any two image sets. As a possible way to tackle these issues, this paper presents a graph embedding multi-kernel metric learning (GEMKML) algorithm for image set classification. The proposed GEMKML implements set modeling, feature extraction, and classification in two steps. Firstly, the proposed framework constructs a novel cascaded feature learning architecture on Grassmannian manifold with the aim of producing more effective Grassmannian manifold-valued feature representations. To make a better use of these learned features, a graph embedding multi-kernel metric learning scheme is then devised to map them into a lower-dimensional Euclidean space, where the inter-class distances are maximized and the intra-class distances are minimized. We evaluate the proposed GEMKML on five different visual classification tasks using widely adopted datasets. The extensive classification results confirm its superiority over the state-of-the-art methods.**

***Index Terms*—Image set classification, Grassmannian manifold, Feature extraction, Graph embedding multi-kernel metric learning.**

I. Introduction

RECENTLY, large amount of videos, such as surveillance videos, handheld camera videos, and drive recorder videos, etc., have been recorded. As each video sequence can be treated as an image set, the interest in image set classification has been steadily increasing [1]–[12]. Video-based face recognition [1], [8], [11], [12], object categorization [4], [7], action recognition [9], [13], facial emotion recognition [13] and face verification [8], [12], [14] are some of its practical applications. Comparing to the conventional single-shot image based classification problem, some key distinguishing features of image set classication can easily be identified. Firstly, the goal of image set classification is to classify the whole image set (a video sequence) into one of the given categories, assuming that each set generally consists of a number of images belonging to the same category. Secondly, an image set is more representative of the data variability, which is important from the point of view of video scene parsing.

Among the existing image set models, linear subspaces have shown impressive ability to derive very good feature representation [2], [13]–[15], [17]–[19]. The advantages of using linear subspace for video based set data description include low computational cost and the capacity of accommodating the effects of various intra-set variations. For these characteristics, it is selected to encode each image set in this paper. As well studied in [2], [14], [20], the distinctive geometrical structure spanned by a set of linear subspaces is usually not a vector space, but instead a Riemannian manifold. Specifically, it is a nonlinear Grassmannian manifold. Hence, applying Euclidean learning techniques to perform vector space dissimilarity measurements for Grassmannian manifold-valued data is inappropriate. In order to overcome this problem, [2], [14], [15], [21] exploit the projection mapping to map each Grassmannian manifold-valued element into a flat space, which is generated by endowing the Grassmannian manifold with the widely used Projection Metric (PM) [2], designed for operations on Grassmannian manifold. After the projection, Euclidean learning algorithms can be applied for computation. Alternatively, several works [2], [5], [15] suggest to exploit the Grassmann kernel [2], which is derived by computing the inner product in the flat space, to embed the Grassmannian data into the Reproducing Kernel Hilbert Space (RKHS) where Euclidean geometry applies.

In the past decade, impressive improvements have been made in the accuracy and speed of video-based image set classification [1], [2], [4]–[7], [9], [12]–[15], [22], [23], [28]–[30], [35]. Among these works, the Riemannian manifold based approaches

Manuscript received September 16, 2019; revised February 14, 2020; accepted February 28, 2020. Date of publication March 18, 2020; date of current version December 17, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC1601800, in part by the National Natural Science Foundation of China under Grants 61672265 and U1836218, in part by the 111 Project of Ministry of Education of China under Grant B12018, and in part by the U.K. EPSRC (EP/N007743/1, MURI/EPSRC/DSTL, EP/R018456/1). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. S. Knorr. *(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 of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2020.2981189

have attracted much attention because they are more effectively in modeling the nonlinearity of image set data. To the best of our knowledge, the existing Riemannian manifold image set classification methods can be grouped into four categories: 1) Riemannian kernel based methods [1], [2], [4], [15], [16]; 2) Riemannian manifold dimensionality reduction methods [7], [9], [14]; 3) Multiple statistical features based approaches [5], [6], [22], [23]; 4) Riemannian deep learning methods [13], [29]–[31]. To be specific, the first type of methods attempt to make the Euclidean computations valid for the Riemannian manifold-valued data by adopting well-studied Riemannian kernel functions [2], [4], [16], [23] to perform high-dimensional Hilbert space embedding. The methods are based on the idea of approximate computation and ignore the manifold geometry of the original image set data. To address this problem, some Riemannian manifold dimensionality reduction algorithms directly learn a map from the original high-dimensional Riemannian manifold to a lower-dimensional, more discriminative one, via Riemannian metric learning. Their strength is mainly manifested in preserving the intrinsic geometrical structure of the data in the feature transformation process. However, the inherent problem of mapping a linear function learned on the nonlinear manifold may cause classification performance degradation.

As a matter of fact, the approaches mentioned above just model each given image set from a single geometric perspective, which may lose some other useful information for classification, especially in the case of complex video data. To tackle this problem, several works [5], [6], [22], [23] attempt to encode each given image set in terms of multiple statistics such as the first-order, second-order and high-order statistics, for the purpose of obtaining complementary feature information. Since different statistics lie in different topological spaces, the metric learning framework is then devised to merge these heterogeneous feature representations into a lower-dimensional common subspace for improved classification. Nevertheless, due to illumination, pose, expression and changes of other conditions in the video capturing process, it is challenging for these approaches to faithfully characterize the real structure of image set data as they are very likely to exhibit large within-class variations. This raises another challenge, namely how to effectively measure the similarity of any two image sets so as to make the intra-class compactness and the inter-class separability of the learned features enhanced as much as possible.

In parallel with the above developments, deep neural networks have become a vital tool in the field of computer vision and pattern recognition. Their advantages stem both from the ability to extract powerful feature representations and from the effective non-linear backpropagation computation [37]–[40] to accomplish learning. Inspired by these merits, some authors embarked on applying the idea of conventional deep learning to Riemannian manifolds to open up a new direction for video-based image set classification. They proposed a spectrum of architectures [13], [29], [31] to perform nonlinear deep feature learning on the Riemannian manifolds. The idea has been successfully applied to action recognition [13], [29], facial emotion recognition [13], [29], and hand action recognition [31], mainly due to the following two factors: 1) the Riemannian geometry of the input data is fully preserved under the end-to-end learning framework; 2) the proposed nonlinear deep learning mechanisms retain the fine-grained semantic information.

To cope with the above mentioned two issues, we propose a graph embedding multi-kernel metric learning (GEMKML) algorithm to learn a powerful feature representation for improved classification. Inspired by the proven success of Riemannian manifold deep learning algorithms, this GEMKML framework first builds a cascaded feature extraction architecture (CasArct) for the purpose of generating more efficient and compact Grassmannian manifold-valued features. It resembles the principle component analysis (PCA) for unsupervised filter learning, rather than the extremely popular Riemannian matrix backpropagation computation, which makes the design of the system and its training much simpler. It also constructs a projection pooling layer by utilizing a two-directional two-dimensional principal component analysis ($(2D)^2$PCA) technique, which not only compacts the generated Grassmannian manifold-valued features, but also maintain the Grassmannian geometry of the input data. To mine the new features better, we first employ the Grassmannian kernel function to embed them into the explicit kernel Hilbert space, inspired by [5], [23]. Then, a graph embedding mechanism-guided multi-kernel metric learning framework is developed to transform the kernel features into a lower dimensional subspace, where the discriminatory power of the resulting subspace features is expected to be enhanced. Our motivation is two-fold: 1) The proven success of the graph-embedding schemes in discovering both the local geometrical structure and discriminative information of image set data [15], [24], [25]; 2) The capability of multi-kernel embedding to learn adaptive weights for each local kernel region reflecting their importance [22], [26].

In contrast to the existing Grassmannian manifold discriminant analysis approaches, which implement feature embedding and classification by taking the path (a)-(b)-(c)-(d) or alternatively (a)-(b)-(e), introduced in Fig. 1, our model addresses two issues: 1) the problem of large within-class ambiguity and low inter-class separability; 2) how to effectively measure dissimilarity of two samples, using a novel learning mechanism (see Fig. 1(a)-(b)-(f)-(g)). To be specific, the proposed framework consists of two progressive stages for fine-grained feature learning, which are clearly shown in Fig. 2. In the first stage, a lightweight cascaded feature extractor is designed to hierarchically extract discriminative visual information from the original Grassmannian manifold. Compared with the conventional hand-crafted counterparts, the generated multiple lower-dimensional features are complementary to each other, such that the intra-class variational information is encoded more succinctly. In the second stage, with the help of the constructed graph model and the multi-kernel learning mechanism, our metric learning framework not only simultaneously exploits the holistic and local geometrical structural information, but also takes the contribution of each feature region into consideration in the fine-grained semantic space embedding process. The evaluation of the proposed GEMKML algorithm on five widely used benchmarking datasets demonstrates its effectiveness.

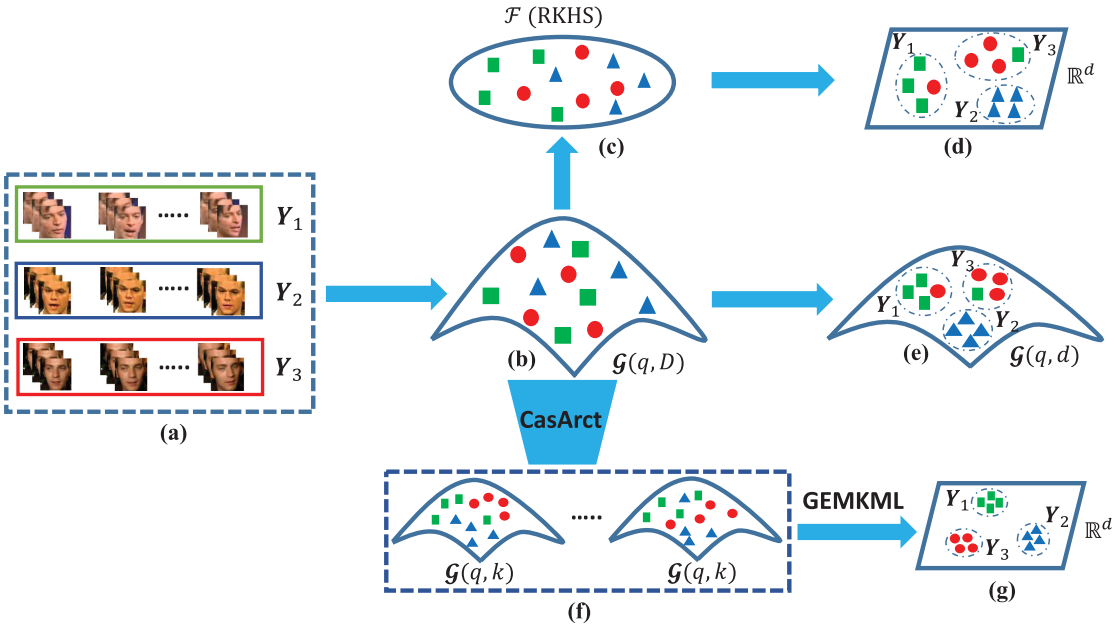


Fig. 1. An intuitive illustration of the proposed graph embedding multi-kernel metric learning (GEMKML) framework. A number of conventional Grassmannian manifold learning algorithms can be used (a)-(b)-(c)-(d) to transform the original Grassmannian manidold $\mathcal{G}(q, D)$ (b) into RKHS (c), then learn a map from RKHS to a discriminative lower-dimensional Euclidean space $\mathbb{R}^d$ (d). The next step is to project the result of (a)-(d) $\mathcal{G}(q, D)$ into a lower-dimensional, but a more powerful counterpart $\mathcal{G}(q, d)$ (e) via a learnable map. In contrast, the proposed approach follows the path (a)-(b)-(f)-(g) first to learn multiple discriminative lower-dimensional Grassmannian manifolds $\mathcal{G}(q, k)$ (f) via the newly designed feature extractor, CasArct. Then a map from the Grassmann manifold-valued feature space to a fine-grained semantic embedding space $\mathbb{R}^d$ (g) is designed so that the within-class compactness and the between-class separation are both enhanced, with the aid of the proposed GEMKML learning scheme.

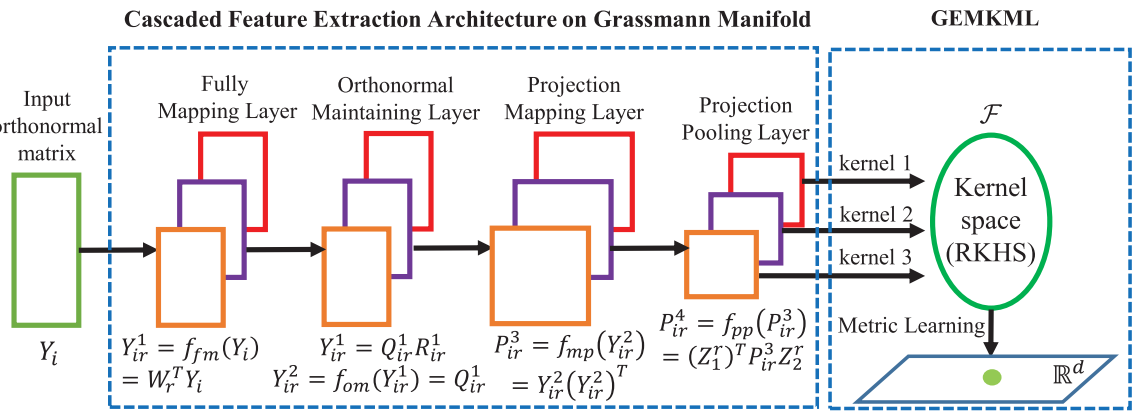


Fig. 2. Details of the architecture of the proposed GEMKML framework. It consists of two subsystems: feature extraction, and image set classification. The first part is constituted by a cascaded feature extractor, which contains a fully connected mapping layer, an orthonormal information preserving layer, a projection mapping layer and a projection pooling layer for the purpose of producing more effective Grassmannian manifold-valued features. The second component is a graph embedding scheme-guided multi-kernel metric learning framework designed to map the features output by the first subsystem into a lower dimensional, but more discriminative common subspace for classification.

II. Background Theory

Before presenting our algorithm, we give a brief overview of the Grassmannian manifold geometry, which provides the foundations for the proposed algorithm.

Given an orthogonal matrix Q of size d-by-d, its equivalence class $[Q]$ can be expressed as follows,

$$[Q] = \left\{ Q \begin{pmatrix} Q_q & 0 \\ 0 & Q_{d-q} \end{pmatrix} : Q_q \in O_q, Q_{d-q} \in O_{d-q} \right\} \quad (1)$$

whose leading q columns form the same subspace as that of Q. Here, O_n is an orthogonal group composed of d-by-d orthogonal matrices. Actually, the equivalence class $[Q]$ represents a point lying in the Grassmannian manifold $\mathcal{G}(q, d) = O_d/(O_q \times O_{d-q})$. In other words, a Grassmannian manifold $\mathcal{G}(q, d)$ is spanned by a set of q-dimensional linear subspaces of $\mathbb{R}^{d \times q}$. Each linear subspace, which is constituted by an orthonormal basis matrix Y of size $d \times q$, ($Y^T Y = I_q$, and I_q is an identity matrix of size $q \times q$), is known as an element of $\mathcal{G}(q, d)$.

As shown in [14], [20], each Grassmannian manifold-valued element corresponds to a unique projection matrix YY^T of size $d \times d$ with rank-q. Accordingly, a natural choice of inner product can be made under a projection operator $\Phi(Y)$ defined as $\langle Y_1, Y_2 \rangle_\Phi = tr(\Phi(Y_1)^T, \Phi(Y_2))$, inducing a geodesic distance named Projection Metric [2],

$$d_{PM}(Y_1Y_1^T, Y_2Y_2^T) = 2^{-1/2}\|Y_1Y_1^T - Y_2Y_2^T\|_{\mathrm{F}} \quad (2)$$

where $\|\cdot\|_{\mathrm{F}}$ indicates the matrix Frobenius norm. Since the projection mapping is continuous and differentiable, the dimensionality of the flat space associated with the Grassmannian manifold can be reduced by endowing it with a Riemannian metric. To compute the inner product in this flat space, we can adopt a Grassmannian kernel

$$k_p(Y_1Y_1^T, Y_2Y_2^T) = tr[(Y_1Y_1^T)(Y_2Y_2^T)] = \|Y_1^TY_2\|_{\mathrm{F}}^2 \quad (3)$$

Its validity has been proven in [2]. Please refer to [20] for the mathematical theory underlying the Grassmannian geometry.

III. Proposed Algorithm

This section presents the proposed method. Section III-A discusses the details of the innovative cascaded feature extraction architecture. We introduce the proposed GEMKML algorithm in Section III-B. The optimization procedure is presented in Section III-C. This is followed by the presentation of the classification algorithm in Section III-D. Section III-E discusses the computational complexity of our GEMKML. Lastly, we elaborate the relationship between the proposed algorithm and the previous works.

A. Cascaded Feature Extraction Architecture for Grassmannian Manifolds

Let $T = [S_1, S_2, \ldots, S_N]$ be the gallery composed of N image sets, with $L = [l_1, l_2, \ldots, l_N] \in \mathbb{R}^{1\times N}$ being the corresponding label vector. In this gallery, $S_i = [s_i^1, s_i^2, \ldots, s_i^{n_i}] \in \mathbb{R}^{d\times n_i}$ is the i-th image set with n_i frames, where $i = 1 \to N$, and $s_i^j \in \mathbb{R}^{d\times 1}$ represents the j-th frame of the i-th image set. From the previous studies [2], [14], [20] we know that, a q-dimensional linear subspace formed by an orthonormal basis $Y_i \in \mathbb{R}^{d\times q}$, s.t. $S_iS_i^T \simeq Y_i\Sigma_iY_i^T$ can be considered to form a Grassmannian manifold-valued feature representation of the i-th image set S_i, where Σ_i and Y_i respectively denote the matrix of q largest eigenvalues and its corresponding eigenvectors.

1) The Mapping Layer: In this paper, the cascaded feature extraction architecture is basically designed to produce more efficient and compact Grassmannian manifold-valued features. Accordingly, we first design a fully connected mapping layer to transform the input data via a mapping f_{fm}, which is formulated as:

$$Y_{ir}^1 = f_{fm}^1(W_r, Y_i) = W_r^TY_i \quad (4)$$

where $Y_i \in G(q, d)$ is the i-th input orthonormal basis matrix of the first layer, $W_r \in \mathbb{R}^{d\times d_1}$ $(d_1 < d)$ is the r-th projection matrix (connection weights) and $Y_{ir}^1 \in \mathbb{R}^{d_1\times q}$ represents the r-th new matrix generated by the first layer. Here, $i = 1 \to N$, $r = 1 \to m_1$, and m_1 denotes the number of feature maps in the first layer.

To learn the connection weights, we first use $T^* = [Y_1, Y_2, \ldots, Y_N] \in \mathbb{R}^{d\times qN}$ to represent the gallery T. Furthermore, we subtract the mean from each orthonormal basis matrix Y_i and obtain its centralized form $\bar{Y_i}$. By making a combination of all the $\bar{Y_i}$, we get

$$X = [\bar{Y_1}, \bar{Y_2}, \ldots, \bar{Y_N}] \in \mathbb{R}^{d\times qN}. \quad (5)$$

Given the above definitions, PCA aims to minimize the following reconstruction error within a group of orthonormal bases, i.e.,

$$\min_{M\in\mathbb{R}^{d\times d_1m_1}} \|X - MM^TX\|_F^2, \text{ s.t. } M^TM = I_{d_1m_1} \quad (6)$$

where $I_{d_1m_1}$ is an identity matrix of size $d_1m_1 \times d_1m_1$, and $d_1m_1 \leq d$. Obviously, the solution to Eq. (6) is made up of d_1m_1 largest eigenvectors of XX^T, such that the filters of the first layer are expressed as:

$$W_r = \mathrm{div}_{d,d_1}(V(XX^T)) \in \mathbb{R}^{d\times d_1},\ r = 1, 2, \ldots, m_1 \quad (7)$$

where $V(XX^T)$ represents the d_1m_1 leading eigenvectors of XX^T, and $\mathrm{div}_{d,d_1}(V)$ is a function that can successively divide $V(XX^T)$ into m_1 parts with each part composed of d_1 eigenvectors.

However, the above learning process also raises a problem which is how to guarantee the orthogonality of each resulting matrix Y_{ir}^1 in the first layer, thus forming a valid Grassmannian manifold.

2) Orthonormal Maintaining Layer: To ensure the underlying space of the resulting matrices is a true Grassmannian manifold, we follow [13] to design a layer to normalize the input matrix Y_{ir}^1 by exploiting QR decomposition, so that the orthogonality of each column of the generated Y_{ir}^2 is preserved. The QR factorization of Y_{ir}^1 can be described as:

$$Y_{ir}^1 = Q_{ir}^1R_{ir}^1 \quad (8)$$

where $Q_{ir}^1 \in \mathbb{R}^{d_1\times q}$ is an orthonormal matrix comprised of q leading columns, and squared matrix R_{ir}^1 is an invertible upper-triangular matrix of size $q \times q$. By utilizing the mathematical properties of Q_{ir}^1 and R_{ir}^1, the input matrix Y_{ir}^1 can be normalized to an orthonormal basis matrix Y_{ir}^2 in the second layer as:

$$Y_{ir}^2 = f_{om}(Y_{ir}^1) = Y_{ir}^1(R_{ir}^1)^{-1} = Q_{ir}^1 \quad (9)$$

where $r = 1 \to m_1$, and $i = 1 \to N$.

3) Projection Mapping Layer: As studied in [14], [20], each Grassmannian point Y of size $d \times q$ uniquely corresponds to a projection matrix $YY^T \in \mathbb{R}^{d\times d}$, and thus can be used to represent Y. Thereby, we design the third layer to perform Grassmannian computing by adopting a projection mapping to each input orthonormal matrix $Y_{ir}^2 \in \mathbb{R}^{d_2\times q}$ with a function f_{pm} formulated as [13]:

$$P_{ir}^3 = f_{mp}(Y_{ir}^2) = Y_{ir}^2(Y_{ir}^2)^T,\ r = 1, 2, \ldots, m_1 \quad (10)$$

where $i = 1 \to N$, and $P_{ir}^3 \in \mathbb{R}^{d_3\times d_3}$ are the resulting projection matrices.

Now, each input data can be represented by m_1 feature maps in this layer. Taking the impact of the dimensionality on information extraction into account, we further design a projection pooling layer to make the resulting data compact.

4) Projection Pooling Layer: It is universally acknowledged that pooling layers with max, min and mean operation functions are often treated as a strategy to reduce the complexity of the ConvNets and further improve the invariance property of the learned features. Without loss of generality, we design the fourth layer to design a pooling function for these learned Grassmannian feature representations, aided by the (2D)^{2}PCA algorithm. This pooling function f_{pp} is expressed as.

$$P_{ir}^4 = f_{pp}(Z_1^r, Z_2^r, P_{ir}^3) = (Z_1^r)^T P_{ir}^3 Z_2^r \tag{11}$$

where $r = 1 \to m_1$, $i = 1 \to N$, $P_{ir}^4 \in \mathbb{R}^{d_4 \times d_4}$ are the generated lower-dimensional and more compact new projection matrices, and $Z_1^r \in \mathbb{R}^{d_3 \times d_4}$ and $Z_2^r \in \mathbb{R}^{d_3 \times d_4}$ ($d_4 < d_3$) are the two directional transformation matrices to be learned.

To make (2D)^{2}PCA play a role, we first treat each input symmetric positive semi-definite (SPSD) projection matrix of the fourth layer as a basic sample. Then, we formulate the sample covariance matrix as:

$$C = \frac{1}{Nm_1 - 1} \sum_{i=1}^{N} \sum_{r=1}^{m_1} \bar{P_{ir}^3} (\bar{P_{ir}^3})^T \tag{12}$$

where $\bar{P_{ir}^3}$ denotes the centralized form of P_{ir}^3, and $C \in \mathbb{R}^{d_3 \times d_3}$ is computed from the column direction (it is equal to the row direction). 2D^{2}PCA maximizes the following criterion to learn the mapping $A \in \mathbb{R}^{d_3 \times d_4 m_1}$,

$$A^* = \arg\max_A J(A), \text{s.t. } A^T A = I_{d_4 m_1} \tag{13}$$

where $J(A) = A^T C A$. In fact, this is an eigenvalue problem and its solution is defined by a family of eigenvectors corresponding to the $d_4 m_1$ largest eigenvalues of C. Finally, each transformation matrix Z_1^r (here, Z_1^r is equal to Z_2^r) can be described as:

$$Z_1^r = \mathrm{div}_{d_3,d_4}(A) \in \mathbb{R}^{d_3 \times d_4},\ r = 1, 2, \ldots, m_1 \tag{14}$$

where the function $\mathrm{div}_{d_3,d_4}(A)$ plays the same role as introduced in Eq. (7).

B. Graph Embedding Multi-Kernel Metric Learning

Now, we use $H^r = [H_1^r, H_2^r, \ldots, H_N^r]$ to denote the resulting r-th feature maps of the gallery. $H_i^r \in \mathbb{R}^{d_4 \times d_4}$ is equivalent to P_{ir}^4, which indicates the r-th feature map is extracted from S_i. Here, $r = 1 \to m_1$, and we set $m_1 = 3$. Currently, a challenge for us is how to make an efficient integration of such m_1 feature maps to mine more discriminative information for classification. In the light of the proven success of kernel learning [26], we first use the Grassmannian kernel to map all the Grassmannian feature maps into a high dimensional Reproducing Kernel Hilbert Space (RKHS), and then compute the dot product in it. Here, we use ϕ_i^r to represent the generated high dimensional feature of H_i^r, and $\phi : \mathbb{R}^{d_4} \to \mathcal{F}$ is the nonlinear mapping function, where $\mathcal{F}$ indicates the transformed kernel space and $\mathbb{R}^{d_4}$ is the original feature space. Though ϕ is usually implicit, we omit it for simplicity. At this point, the distance between any two given image sets S_i and S_j in $\mathcal{F}$ is defined as:

$$d(S_i, S_j) = tr\left[\sum_{r=1}^{m_1} \rho_r(\phi_i^r)(\phi_i^r - \phi_j^r)^T E(\phi_i^r - \phi_j^r)\rho_r(\phi_j^r)\right] \tag{15}$$

where $\rho_r(\phi_i^r)$ is a gating model defined to allocate different positive weights to different ϕ_i^r that will be introduced later, and E is the Mahalanobis matrix to be learned. Due to the symmetric positive semi-definite property of E, we seek a non-square matrix $B = [b_1, b_2, \ldots, b_{d_b}]$ to reconstruct it as $E = BB^T$. Eq. (15) can therefore be rewritten as.

$$\begin{aligned} &d(S_i, S_j) \\ &= tr\left[B^T \left(\sum_{r=1}^{m_1} \rho_r(\phi_i^r)(\phi_i^r - \phi_j^r)(\phi_i^r - \phi_j^r)^T \rho_r(\phi_j^r)\right) B\right] \end{aligned} \tag{16}$$

As shown in Fig. 2, our target is to learn the Mahalanobis matrix, which is reduced to learning the transformation matrix B to map the Hilbert space data into a common subspace. In this subspace the between-class variations are expected to be maximized and the within-class variations are expected to be shrunk, simultaneously. Inspired by the general graph embedding framework [15], [24], [25], we first set up a graph to encode the local geometrical structure of the original Grassmannian manifold $\mathcal{G}(q, d)$, which is quantified as the following affinity matrix $\Lambda \in \mathbb{R}^{N \times N}$ computed by making the use of the following intra-class similarity graph G_w and the inter-class similarity graph G_b:

$$G_w(Y_i, Y_j) = \begin{cases} w_{ij}, & \text{if } Y_i \in N_w(Y_j) \text{ or } Y_j \in N_w(Y_i) \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

$$G_b(Y_i, Y_j) = \begin{cases} w_{ij}, & \text{if } Y_i \in N_b(Y_j) \text{ or } Y_j \in N_b(Y_i) \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where $w_{ij} = \exp(-d_{PM}^2(Y_i Y_i^T, Y_j Y_j^T)/\sigma)$. Hence, the affinity matrix Λ is defined as: $\Lambda = G_w - \alpha G_b$, where each entry $\Lambda(Y_i, Y_j)$ measures the notion of similarity between Y_i and Y_j, and α is used to balance the compactness of G_w and the dispersion of G_b. In Eq. (17), $N_w(Y_i)$ represents a collection of R_w within-class nearest neighbors of Y_i, while $N_b(Y_i)$ in Eq. (18) indicates a collection of R_b between-class nearest neighbors of Y_i. Note that, we use the Projection Metric (PM) to compute the nearest neighbors. In practice, the value of R_w is configured to the minimum number of samples in a class, and set $R_b \leq R_w$. However, their specific values are determined by cross-validation.

Having obtained this affinity matrix Λ, the objective function of the designed graph embedding multi-kernel metric learning framework is formulated as:

$$B^* = \arg\min_{B \in \mathbb{R}^{N \times d_b}} \Theta(B) = \arg\min_{B \in \mathbb{R}^{N \times d_b}} \sum_{i,j=1}^{N} \Lambda_{i,j} d(S_i, S_j) \tag{19}$$

As it is arduous to compute the distance $d(S_i, S_j)$ defined in Eq. (16) due to the implicity of ϕ, we express the basis b_k as a linear combination of all the training instances in $\mathcal{F}$, i.e., $b_k = \sum_{i=1}^{N} u_i^k \phi_i^r$, where u_i^k are the representation coefficients. Hence, we have.

$$\sum_{r=1}^{m_1} b_k^T \phi_i^r = \sum_{i=1}^{N}\sum_{r=1}^{m_1} u_i^k (\phi_i^r)^T \phi_i^r = \sum_{r=1}^{m_1} (u^k)^T K_{.i}^r \tag{20}$$

where $u^k \in \mathbb{R}^{N\times 1}$ is a column vector and u_i^k is its i-th entry. $K_{.i}^r$ is the i-th column of the r-th kernel matrix K^r. Here, $K^r \in \mathbb{R}^{N\times N}$ is computed from H^r using the Grassmannian kernel. The above Eq. (20) is obtained by using the kernel trick [42].

Therefore, we can reformulate the minimization problem defined in Eq. (19) as,

$$U^* = \underset{U\in\mathbb{R}^{N\times d_b}}{\arg\min}\, \Theta(U) = \underset{U\in\mathbb{R}^{N\times d_b}}{\arg\min} \sum_{i,j=1}^{N} \Lambda_{i,j} d^*(S_i, S_j) \tag{21}$$

where

$$d^*(S_i, S_j) = tr\Bigg[U^T \Bigg(\sum_{r=1}^{m_1} \rho_r(\phi_i^r)(K_{.i}^r - K_{.j}^r) (K_{.i}^r - K_{.j}^r)^T \rho_r(\phi_j^r) \Bigg) U \Bigg] \tag{22}$$

The subject of our next discussion is the gating model $\rho_r(\phi_i^r)$. In this study, its specific form is defined as in [26].

$$\rho_r(\phi_i^r) = \frac{\exp(g_r^T \phi_i^r + h_r)}{\sum_{v=1}^{m_1} \exp(g_v^T \phi_i^v + h_v)} \tag{23}$$

It is clear this model grows incrementally with the importance of ϕ_i^r, and the softmax can guarantee nonnegativity of the resulting weights. Since the form of ϕ is usually unknown, we induce this model to play a part in a similar way to Eq. (17).

$$g_r^T \phi_i^r = \delta_r^T (\phi_i^r)^T \phi_i^r = \delta_r^T K_{.i}^r \tag{24}$$

As a result, this gating model can be rewritten as,

$$\rho_r(\phi_i^r) = \frac{\exp(\delta_r^T K_{.i}^r + h_r)}{\sum_{v=1}^{m_1} \exp(\delta_v^T K_{.i}^v + h_v)} \tag{25}$$

where $\delta_r \in \mathbb{R}^{N\times 1}$ and $h_r \in \mathbb{R}^1$ are the two parameters to be learned.

C. Optimization

Owing to interdependence between U and (g_r, h_r), it is difficult to find a closed-form solution for the optimization problem defined in Eq. (21). With this in mind, we use an iterative procedure to solve it. The basic idea is first to fix the values of δ_r and h_r to update U, and then update δ_r and h_r with the updated U. Finally, this staggered procedure iterates until convergence.

1. *Computation of U:* In order to optimize U, we first initialize δ_r and h_r with a small random vector of size $N \times 1$ and a small random constant respectively, and impose a unitary constraint $U^T U = I_{d_b}$ to U for restricting its scale. Generally speaking, this optimization problem is on the Stiefel manifold [20], [27]. Note, the objective function $\Theta(U)$ is independent of the bases spanned by U since it satisfies $\Theta(U) = \Theta(UQ)$. Here, $Q \in O_{d_b}$, and O_{d_b} are an orthogonal group formed by a family of $d_b \times d_b$ orthogonal matrices as introduced in Section II. As a result, this optimization problem is actually on the Grassmannian manifold [20], [27], and the nonlinear Grassmannian Conjugate Gradient method [9], [20], [27] can be applied for its optimization. The following is a brief summarization of the Conjugate Gradient method on the Grassmann manifold: (1) At the k-th iteration, use the following formula to compute the Grassmannian gradient $\nabla_U \Theta(U_k)$ at current solution:

$$\nabla_U \Theta(U_k) = (I_N - UU^T) D_U \Theta(U_k) \tag{26}$$

where $D_U\Theta(U_k)$ denotes the Euclidean gradient of $\Theta(U_k)$ with respect to U_k, and can be expressed as: $D_U\Theta(U_k) = 2\sum_{i,j=1}^{N} \Lambda_{i,j} \Delta U_k$. Here, the specific form of Δ is given as follows;

$$\Delta = \sum_{r=1}^{m_1} \rho_r(\phi_i^r)(K_{.i}^r - K_{.j}^r)(K_{.i}^r - K_{.j}^r)^T \rho_r(\phi_j^r)) \tag{27}$$

(2) Determine the new search direction Γ_k by integrating the parallel update using $\tau(\Gamma_{k-1}, U_{k-1}, U_k)$ [9] of the previous search direction Γ_{k-1} from U_{k-1} to U_k with $\nabla_U \Theta(U_k)$. In other words $\Gamma_k \leftarrow -\nabla_U \Theta(U_k) + \lambda\tau(\Gamma_{k-1}, U_{k-1}, U_k)$; (3) Find $U_k = \arg\min \Theta(U)$ based on line search along the geodesic on the Grassmannian manifold $\mathcal{G}$ from U_k with direction Γ_k.

Then, repeat the above steps until one of the following conditions is satisfied: 1) the objective function $\Theta(U)$ converges to a local minimum; 2) reaches the maximum iterations. For a more comprehensive introduction of this optimization algorithm, please refer to [20], [27].

2. *Computation of δ_r and h_r:* After updating U, we then respectively compute the partial derivatives of $\Theta(U)$ with respect to δ_r and h_r as:

$$\begin{aligned}\frac{\partial\Theta(U)}{\partial\delta_r} &= UU^T \sum_{i,j=1}^{N} \Lambda_{i,j} \frac{\partial\Delta(\delta_r, h_r)}{\partial\delta_r} \\ &= UU^T \sum_{i,j=1}^{N}\sum_{k=1}^{m_1} \Lambda_{i,j}\rho_k(\phi_i^k)(K_{.i}^k - K_{.j}^k) \\ &\quad (K_{.i}^k - K_{.j}^k)^T \rho_k(\phi_j^k)[K_{.i}^r(\beta_r^k - \rho_r(\phi_i^r)) \\ &\quad + K_{.j}^r(\beta_r^k - \rho_r(\phi_j^r))] \end{aligned} \tag{28}$$

$$\begin{aligned}\frac{\partial\Theta(U)}{\partial h_r} &= UU^T \sum_{i,j=1}^{N} \Lambda_{i,j} \frac{\partial\Delta(\delta_r, h_r)}{\partial h_r} \\ &= UU^T \sum_{i,j=1}^{N}\sum_{k=1}^{m_1} \Lambda_{i,j}\rho_k(\phi_i^k)(K_{.i}^k - K_{.j}^k) \\ &\quad (K_{.i}^k - K_{.j}^k)^T \rho_k(\phi_j^k)\Big[\beta_r^k - \rho_r(\phi_i^r) \\ &\quad + \beta_r^k - \rho_r(\phi_j^r)\Big] \end{aligned} \tag{29}$$

Algorithm 1: GEMKML

Input: Training image sets T, label vector L, m_1 different kernel matrices K^r ($r = 1 \rightarrow m_1$), number of iterations G, target dimensionality d_b, affinity matrix Λ and convergence error ε.

Output: Transformation matrix U and parameters δ_r, h_r.

Step 1 (Initialization): Initialize δ_r^0 with a small random vector, and initialize h_r^0 with a small random constant.

Step 2 (Optimization):

For $i = 1, 2, \ldots, B$, repeat

1: Solve the minimization problem in Eq. (21) using Grassmannian Conjugate Gradient method, and get $U^i = [u_1, u_2, \ldots, u_{d_b}]$.

2: Compute $\frac{\partial \Theta(U)}{\partial \delta_r}$ and $\frac{\partial \Theta(U)}{\partial h_r}$ using Eq. (28) and Eq. (29), respectively.

3: Update δ_r and h_r using Eq. (30) and Eq. (31), respectively.

4: Check convergence:

if $i > 2, |\delta_r^{(i+1)} - \delta_r^i| < \varepsilon$ and $|h_r^{(i+1)} - h_r^i| < \varepsilon$,

or $|U^{i+1} - U^i| < \varepsilon$, turn to Step 3.

Step 3 (Output): Transformation matrix U, δ_r and h_r

where $\beta_r^k = 1$ if $r = k$ and 0 otherwise. Here, the gradient descent method is applied to train the gating model defined in Eq. (25) as.

$$\delta_r^{t+1} = \delta_r^t - \eta \frac{\partial \Theta(U)}{\partial \delta_r} \quad \text{and} \quad h_r^{t+1} = h_r^t - \eta \frac{\partial \Theta(U)}{\partial h_r} \tag{30}$$

where η is the learning rate and set to 10^{-3} in our experiments.

Having obtained the new δ_r and h_r, we first use them to recalculate $\rho_r(\phi_i^r)$ and Δ in Eq. (22) and Eq. (27), respectively. Then, the transformation matrix U can be updated by reusing the Grassmannian Conjugate Gradient method. Finally, we iterate this alternating process until the conditions are met. We summarize the proposed GEMKML algorithm in Algorithm 1.

D. Classification

In the test stage, the proposed cascaded feature extraction architecture (CasArct) is firstly exploited to process a given test image set S_{te}, and its m_1 corresponding feature maps can therefore be produced, expressed as H_{te}^r, and $r = 1 \rightarrow m_1$. Then, we measure the similarity between S_{te} and all the training image sets using three kernel vectors calculated via the Grassmannian kernel, each represented by $K_{.te}^r$. Therefore, the distance between S_{te} and each training image set S_i can be computed as follows:

$$d(S_{te}, S_i) = tr\left[\left(\sum_{r=1}^{m_1} \rho_r(\phi_{te}^r)(K_{.te}^r - K_{.i}^r)UU^T (K_{.te}^r - K_{.i}^r)^T \rho_r(\phi_i^r)\right)\right] \tag{31}$$

Lastly, we assign label l_i to S_{te} according to:

$$l_i = \arg\min_i d(S_{te}, S_i) \tag{32}$$

where $i = 1 \rightarrow N$.

E. Computational Complexity Analysis

The computational complexity in the training stage is determined by the following six factors: 1) the cost $\mathcal{O}(d^2qN + d^3)$ of performing the PCA learning in the fully connected mapping layer; 2) the cost $\mathcal{O}(Nq^2d_1)$ of QR decomposition in the orthonormal maintaining layer; 3) performing the pooling operation in the projection mapping layer, which requires $\mathcal{O}(2d_3^3 m_1 N + 2d_3^2 d_4 m_1 N)$ operations; 4) building m_1 kernel matrices, which costs $\mathcal{O}(m_1 N^2 d_4^3)$; 5) constructing the graph, which requires $\mathcal{O}(N^2(d^2q + d^3))$ operations; 6) the cost $\mathcal{O}(2Jm_1 N^2)$ for updating δ_r and h_r. Here, J represents the number of iterations. Considering that $m_1 \ll N$, $J \ll N^2$, $d_1 \leq d$, $d_3 \leq d_1$, and $d_4 \leq d_3$, the computational complexity of our algorithm is $\mathcal{O}(N^2\, d^3)$.

F. Relationship With the Previous Works

Our method is similar to [13], [16], [23]. Here, we point out some essential differences between the proposed GEMKML and those introduced in [13], [16], [23] in the following three paragraphs.

[13] is a Grassmannian deep learning network, which consists of some elaborately designed spectral layers for extracting Grassmannian manifold-valued features using deep network. Our designed CasArct is motivated by [13], but the differences lie in two respects: 1) we use PCA to perform unsupervised learning, which means the training of CasArct is extremely easy and efficient, while [13] depends on end-to-end learning with a Riemannian matrix backpropagation computation; 2) We design a projection pooling layer by using (2D)^{2}PCA technique, which not only refines the CasArct but also preserves the Riemannian geometrical structure of the data. [13] exploits a mean pooling operation for Grassmannian data.

Both the proposed model and [16] learn discriminative Euclidean feature representations for Riemannian manifold-valued data. Specifically, they first embed the Riemannian manifold features into RKHS, and then learn a map from the explicit kernel spaces into a lower-dimensional Euclidean space where the discriminatory power is enhanced. However, they have some essential differences in the mechanism of representation learning. Firstly, [16] extracts the Grassmann manifold-valued features for the ETH-80 dataset without any preprocessing steps while the proposed method utilizes the newly designed CasArct to learn hierarchical Grassmannian features. Secondly, the multiple explicit kernel spaces of [16] are generated by different types of Grassmannian kernels. For the proposed approach, only m_1 projection kernels are used to perform the explicit kernel space transformation. Thirdly, [16] considers the linear combinations of the heterogeneous explicit kernel feature representations, while our algorithm works with kernel combinations in the multi-kernel learning scheme. Finally, [16] formulates the

discriminative feature representation learning problem in the framework of the SVM theory. In contrast, the graph embedding mechanism guided objective function is used to solve this problem in the proposed model. Note that, we do not select [16] as the comparative method in the following experiments because it uses one-vs-one scheme for multi-class classification problems, which differs from the proposed algorithm.

Our approach and [23] not only focus on building reliable image set models, but also on learning discriminative subspace features. However, there are some differences between the two approahces: 1) We use the proposed CasArct to automatically learn the features, while [23] relies on the multiple statistics based hand-crafted ones; 2) As the discriminability of each local region in the resulting kernel spaces is different, this paper integrates the multi-kernel learning scheme into the proposed metric learning framework with the aim of learning adaptive weight for each kernel. These differences are ignored in [23]; 3) Regarding optimization, this paper first formulates the feature fusion problem using an elaborately designed graph embedding mechanism-guided objective function, and then exploits RCG [20], [27] and gradient descent method to solve it iteratively. [23] utilizes the LogDet divergence [32] based constraint to formulate the feature fusion problem, and solves it with the cyclic Bregman projection method [33]; 4) In this paper we evaluate the proposed algorithm on five different video-based classification tasks, and the extensive classification results demonstrate its wider promise, while [23] concentrates on video-based face recognition task.

IV. Experiments

In this section, we assess the proposed GEMKML algorithm[1] on four classification tasks, e.g., video-based face recognition, video-based emotion recognition, dynamic scene classification and set-based object categorization, respectively.

A. Comparative Methods and Settings

We compare the proposed GEMKML algorithm with some representative image set classification methods including: Grassmann Discriminant Analysis (GDA) [2], Grassmannian Graph-Embedding Discriminant Analysis (GEDA) [15], Covariance Discriminative Learning (CDL) [4], Riemannian Sparse Representation (RSR) [28], Projection Metric Learning (PML) [14], SPD Manifold Learning (SPDML) Based on Affine-Invariant Metric (AIM) [9] and Stein Divergence [9], Log-Euclidean Metric Learning (LEML) [7], Hybrid Euclidean-and-Riemannian Metric Learning (HERML) [23], Localized Multi-kernel Metric Learning (LMKML) [22], Symmetric Positive Definite (SPD) Network (SPDNet) [29], and Grassmannian manifold Network (GrNet) [13].

We use the source codes of all the comparative methods provided by the original authors to conduct the experiments, except for LMKML. Since the source code of LMKML is not available, we carefully implement it by referring to [22]. For a fair comparison, the parameters of all the methods were set in this paper are empirically tuned according to the information provided in the original works. For CDL, the perturbation is set to $10^{-3} \times trace(C)$. For GDA and GEDA, the number of basis vectors for the subspace is determined by cross-validation. Moreover, KDA is utilized to perform discriminative learning in CDL and GDA. In PML, the number of iterations, the trade-off coefficient α and the target dimensionality d are reported according to the original work [14]. In LEML, we search the values of η and ζ in the range of [0.1,1,10] and $[0.1 : 0.1 : 1]$, respectively. In SPDNet, the number of iterations and the size of input data are respectively set to 500 and 400×400, while they are configured as 500 and 400×10 in GrNet. Other key parameters in SPDNet and GrNet such as learning rate, batch size and the size of the transformation matrices are chosen by cross-validation. For RSR, the value of λ is tuned by sampling the range [0.0001,0.001,0.01,0.1]. For SPDML-AIM and SPDML-Stein, v_w and v_b are searched by referring to [9], while the target dimensionality of the resulted new SPD manifold is set by cross-validation. In HERML, we respectively tune γ and ζ in the range of [0.001,0.01,0.1,1,10,100,100] and $[0.1 : 0.1 : 1]$. In LMKML, the learning rate α is set as 10^{-6}. Note that, for the parameters determined by cross-validation, we report the best classification results for such methods in this paper.

TABLE I
The Suitable Values of Some Fundamental GEMKML Parameters

Dataset	q	d_1	d_4	m_1	R_w	R_b	α	σ	d_b
YTC	11	90	30	3	3	2	0.5	5	70
AFEW	16	90	30	3	16	16	0.5	4	70
MDSD	15	90	30	3	5	5	0.5	5	25
ETH-80	10	85	28	3	5	3	0.2	4	10

To build the cascaded feature extraction architecture, we use four layers: $Y_i \rightarrow f_{fm} \rightarrow f_{om} \rightarrow f_{mp} \rightarrow f_{pp}$, shown in Fig. 2. The recommended values of parameters q, d_1, d_4, and m_1, which are closely related to this architecture, are given in Table I. Here, q represents the dimensionality of the original Grassmann manifold, d_1 and d_4 respectively denote the dimensionality of the produced feature maps in the fully connected mapping layer and the projection pooling layer, and m_1 is the number of feature maps in each layer. Note that, they are determined by cross-validation, and we suggest to exploit fine tuning to choose suitable values for new problems.

B. Datasets Description and Settings

In our experiments, the challenging and widely used YouTube Celebrities (YTC) dataset [4], [5], [7], [23] is adopted for the task of video-based face recognition. It contains 1,910 video clips of 47 subjects. Each clip is comprised of hundreds of face frames, most of which exhibits large intra-class variations in expression, pose, resolution and illumination. Some sample face images of this dataset are shown in Fig. 3, and histogram equalization is adopted to each face region of this dataset for eliminating the lighting effects.

Dynamic scene classification in an unconstrained setting is a fundamental and challenging task in computer vision. Recently, image set classification has provided a new direction to address

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

Fig. 3. Face frames of the YTC dataset.

Fig. 4. Dynamic scene images of the MDSD dataset.

Fig. 5. Examples of the ETH-80 dataset.

Fig. 6. Facial emotion images of the AFEW dataset.

this problem. In this paper, we use the MDSD dataset [35], [36], [41] to evaluate the classification performance of the proposed method on this task. As presented in Fig. 4, there are some sample images of this dataset. This dataset is composed of 13 different categories of dynamic scenes. Each has 10 video sequences collected in an unconstrained setting, and they exhibit large within-class diversity in resolution and morphology.

For the task of set-based object categorization, we use the ETH-80 dataset [4], [5], [7], [22]. It consists of 8 classes such as apples, cows, cups, dogs, horses, pears, tomatoes and cars, with 10 subcategories per class, and each subcategory contains 41 instances collected from different perspectives. Fig. 5 shows some examples of this dataset.

We further apply the proposed model to the emotion recognition task using the Acted Facial Expression in Wild (AFEW) dataset [13], [29], [34]. This dataset involves 1,345 video sequences of facial expressions collected from movies with close to real world scenarios. Some examples of the AFEW dataset are presented in Fig. 6. For evaluation, we follow the standard protocols of EmotiW2014 [34] and [13], [29] to split these training video sequences into 1,746 small clips for data augmentation, and report the recognition results on the validation set, as the groundtruth of test set is not publicly available.

To keep consistent with the previous works [4], [5], [7], [13], [14], [23], we conduct ten-fold cross validation experiments, i.e., randomly split each dataset into ten different pairs of gallery and probes, and report the average classification accuracies of each method on the YTC, MDSD and ETH-80 datasets. In particular, for the YTC dataset, we randomly choose 9 video clips in each subject with 3 for training and 6 for testing. For the MDSD dataset, each class has 7 randomly selected videos for training and the rest for the query set. For the ETH-80 dataset, each category has 5 randomly selected objects for gallery and the remaining five for probes. Besides, each image in a given dataset is normalized to a 20×20 grayscale one.

TABLE II
AVERAGE RECOGNITION SCORES (%) OF DIFFERENT METHODS ON THE YTC AND AFEW DATASETS

Methods	YTC	AFEW	MDSD	ETH-80
GDA [2]	66.15	29.11	30.51	93.25
GEDA [15]	66.57	29.45	30.37	94.50
CDL [4]	68.76	31.81	30.51	93.75
RSR [28]	72.77	27.49	31.62	93.25
LMKML [22]	70.31	-	31.74	92.50
HERML [23]	73.28	32.14	32.37	95.00
PML [14]	67.62	28.98	29.32	93.25
LEML [7]	69.04	25.13	29.74	94.00
SPDML-AIM [9]	64.66	26.72	31.10	90.75
SPDML-Stein [9]	61.57	24.55	29.81	90.50
GrNet [13]	70.46	34.23	31.25	91.75
SPDNet [29]	69.38	34.23	32.05	90.25
GEMKML	**74.81**	**35.71**	**35.89**	**97.00**

TABLE III
INVESTIGATION OF THE PROPOSED GEMKML WITH DIFFERENT COMPONENTS

Methods	ETH-80	MDSD	YTC	AFEW
CasArct	94.50	33.85	72.84	32.14
GEMKML-woCA	95.00	31.28	73.83	33.53
GEMKML-womkl	97.00	34.87	74.72	34.40
GEMKML	97.00	35.89	74.81	35.71

Baseline methods are CasArct and GEMKML-woCA. Here, 'woCA' represents without utilizing CasArct, and 'womkl' indicates without utilizing multi-kernel learning.

C. Results and Discussions

The classification results of all the methods on the four datasets are tabulated in Table II and Table III. We can make some interesting observations about these classification results. Firstly, the recognition scores of GDA are inferior to that of GEDA on the YTC, AFEW and ETH-80 datasets, which demonstrates the capability of the graph embedding scheme to extract discriminative local structural information when performing manifold discriminant analysis. Furthermore, it is evident that the recognition rates of GDA and GEDA are both lower than that of PML on the YTC and ETH-80 datasets. The main reason is that PML performs dimensionality reduction by jointly learning an embedding and a distance metric from the original Grassmannian manifold, which retains the manifold property of the original set data more comprehensively than the Euclidean computation. Consequently, more powerful geometrical structural features are mined for improved classification. This reason also explains the difference in the recognition ability between CDL and LEML on the YTC and ETH-80 datasets. Since PML and LEML depend on restricting the intra-class and inter-class dispersion to learn the discriminative distance metrics, it is impossible for them to make effective distinctions between some overlapping samples when there exists large within-class variations within the dataset. This may explain why the classification results of PML and LEML are respectively a bit lower than that of GEDA and CDL on the AFEW and MDSD datasets, which

in turn further verifies the effectiveness of the proposed graph embedding mechanism.

Secondly, we also want to discuss the differences between the results produced by SPDNet, GrNet, LEML, and PML on the four datasets. It is notable that both SPDNet and GrNet show relatively poor classification results than LEML, PML and other image set classification methods on the ETH-80 dataset, while on the relatively large scale AFEW and YTC datasets, they are superior to LEML, PML, and other competitors. Meanwhile, SPDNet and GrNet also achieve comparable classification performance on the MDSD dataset, compared to the state-of-the-art methods. As aforementioned, these four algorithms all aim at transforming the high dimensional Riemannian manifolds into the lower dimensional and more discriminative ones, such that the geometrical structure of the input data is preserved. However, both SPDNet and GrNet depend on the end-to-end learning framework to produce powerful Riemannian manifold-valued feature representations. As a consequence, SPDNet and GrNet are capable of parsing the manifold geometry more effectively thanks to the deep features than other algorithms, and thus show good classification ability on the complicated datasets. For the ETH-80 dataset, its limited number of training samples may be the root cause of the poor classification performance of SPDNet and GrNet.

Thirdly, the comparison of classification performance of LMKML, HERML, and the proposed GEMKML is of the main interest. For HERML and LMKML, their similarities emanate from two roots: 1) they both utilize multiple statistics to simultaneously encode the image set data; 2) to handle heterogeneous feature representations, they exploit the advocated metric learning frameworks to merge them into a unified subspace for classification. It is easy to see that LMKML and HERML outperform most of the comparative methods on the four test datasets, which proves the complementarity of multiple statistics in image set modeling helps to extract more discriminatory information than single geometric model based methods. However, LMKML is convincingly surpassed by HERML. The fundamental reason is that LMKML applies the Euclidean metric based kernel function to the higher order statistics such as second-order statistics and tensors, which does not preserve the Riemannian geometry of the original image set data as they typically reside in a non-Euclidean space. On the contrary, HERML treats different statistics with different kernel functions. As LMKML is very time consuming, we did not experiment with it on the large scale AFEW dataset. For HERML, as discussed before, the lack of making quantitative distinction between different local kernel regions is regarded as the main factor limiting its classification performance.

It is clear to see that the proposed approach achieves state-of-the-art classification performance on all the datasets. This is attributed to the following factors:

1) The proposed cascaded feature extraction architecture (CasArct) can not only reduce the dimensionality of the input data, but also refine the learned Grassmannian manifold-valued features via the novel pooling operation. Consequently, the output of CasArct is more discriminative than the hand-crafted features.

TABLE IV
INVESTIGATION OF DIFFERENT SUBARCHITECTURES OF THE PROPOSED CASACDED FEATURE EXTRACTOR (CASARCT) ON THE ETH-80, MDSD, AND YTC DATASETS

Methods	ETH-80	MDSD	YTC
CasArct-f_{fm}	63.00	18.46	27.59
CasArct-$f_{fm} + f_{om}$	73.00	19.49	41.70
CasArct-$f_{fm} + f_{om} + f_{pm}$	91.25	31.79	71.91
CasArct-$f_{fm} + f_{om} + f_{pm} + f_{pp}$	94.50	33.85	72.84

2) The proposed within-class similarity graph G_w and the between-class similarity graph G_b successfully mine the local structural information of the image set data.
3) With the help of the proposed multi-kernel learning scheme and the graph model, the metric learning framework developed has the capacity to fuse these learned geometric features into a lower dimensional metric space, where the intra-subject distances are minimized and the inter-subject distances are maximized.

D. Ablation Study of the Proposed GEMKML Algorithm

The above experimental results demonstrate the effectiveness of our algorithm for image set classification. To better evaluate its properties, we study two baselines: 1) using just the cascaded feature extractor (CasArct) to perform image set classification; 2) Excluding the CasArct from the proposed GEMKML framework, we name it GEMKML-woCA. First we provide some implementation details of CasArct and GEMKML-woCA: 1) to complete the classification procedures of CasArct, we first vectorize each output of the projection pooling layer, then concatenate them together. Finally, they are fed into the Nearest Neighbor (NN) classifier for classification; 2) To complete the learning process of GEMKML-woCA, the Grassmannian manifold kernel is first applied to the input orthonormal matrix Y_i for Hilbert space embedding. Then, the proposed metric learning framework is utilized to carry out the discriminant subspace learning. The classification accuracies obtained for the two baselines on the four datasets are listed in Table IV. To assess the merits of the multi-kernel learning scheme of the proposed GEMKML, we further make experiments on the four datasets to investigate the classification performance of our GEMKML without using multi-kernel learning mechanism (GEMKML-womkl). The classification results are given in Table IV. Distinctly, both CasArct and GEMKML-woCA achieve is at least comparable, or better on all the datasets, compared to the competitors listed in Table II and Table III. The classification performance of GEMKML-womkl is inferior to the proposed GEMKML on the MDSD, YTC and AFEW datasets, while it surpasses the two baselines. The above observations support the following three point: 1) the effectiveness of the proposed CasArct in producing more efficient and compact Grassmannian manifold-valued features; 2) the validity of the multi-kernel learning scheme in alleviating the issue of within-class variations; 3) The poposed GEMKML framework, which is organically integrated with CasArct, can mine more powerful structured semantic information for improved classification.

TABLE V
AVERAGE COMPUTATION TIME (SECONDS) OF REPRESENTATIVE IMAGE SET CLASSIFICATION METHODS ON THE YTC DATASET

Methods	Training	Testing
CDL [4]	15.75	0.11
GDA [2]	17.55	0.13
PML [14]	660.89	0.05
LEML [7]	192.57	0.61
HERML [23]	26.13	0.15
SPDML-AIM [9]	871.62	1.25
SPDML-Stein [9]	332.73	0.62
SPDNet [29]	1668.06	0.01
GrNet [13]	4680.15	0.03
CasArct	1.51	0.001
GEMKML	40.81	0.06

Note that the testing time of each method is computed by measuring the time required to classify one image set (one video sequence) into the given categories.

E. Ablation Study for the Proposed Cascaded Feature Extractor (CasArct)

As aforementioned, the proposed lightweight cascaded feature extractor (CasArct) produces more efficient and compact Grassmannian manifold-valued feature representations for the image set data. In Section IV-D, we conducted experiments to demonstrate that the classification ability of CasArct is competitive with most of the comparative methods. In order to investigate the importance of each layer in CasArct, we perform classification experiments on the ETH-80, MDSD, and YTC datasets, respectively. The experimental results obtained by different subarchitectures of the suggested CasArct are presented in Table V. From the table, we can find that the classification score of CasArct-f_{fm}, which just includes the fully connected mapping layer, is much lower on the three datasets. The reasons are two-fold: 1) the Grassmannian geometry of the generated features have been destroyed; 2) the features produced contain more redundancy. However, when the orthonormal structural information preserving layer is coupled with the fully connected mapping layer, the classification ability of CasArct-$f_{fm} + f_{om}$ is improved. This demonstrates the importance of preserving the geometrical structure of the transformed data in image set classification. On the basis of CasArct-$f_{fm} + f_{om}$, the projection mapping layer is integrated into the tail of the orthonormal structural information preserving layer. As a result of this measure, the classification performance of CasArct-$f_{fm} + f_{om} + f_{pm}$ is lifted significantly. This suggests that endowing the extracted Grassmannian manifold-valued features with Grassmannian manifold computing enhances the discriminatory structural information content of the extracted features. From Table V, we can also note that when the projection pooling layer is incorporated with CasArct-$f_{fm} + f_{om} + f_{pp}$, the recognition ability of CasArct-$f_{fm} + f_{om} + f_{pm} + f_{pp}$ is further improved on three datasets used in the experimental study. This confirms the effectiveness of the designed pooling layer in producing an efficient and compact feature representation. The above observations demonstrate the significance of each CasArct layer for image set classification.

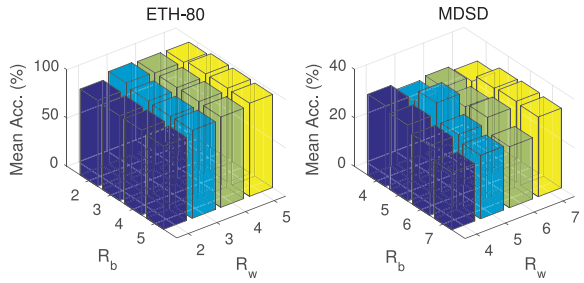


Fig. 7. Mean classification results of GEMKML on the ETH-80 and MDSD datasets under different parameter settings. The values of R_w and R_b are varied in the range of $\{2, 3, 4, 5\}$ and $\{4, 5, 6, 7\}$ respectively.

F. Ablation Study for the Graph Model

Graph embedding scheme has been shown to be effective in making use of the local geometrical structure of image set data. In this context, choosing suitable values for R_w and R_b, which respectively represent the number of within-class and between-class nearest neighbors of Y_i, is instrumental to improving the discriminability of the extracted subspace features. When constructing the graph model, the problem of data imbalance is often encountered because the number of similar sample pairs is much lower than that of dissimilar sample pairs. To mitigate the impact of this problem on the classification performance, it is important to set proper values for R_w and R_b. With this motivation, we conducted experiments on the ETH-80 and MDSD datasets to study the influence of R_w and R_b on the classification performance of the proposed method. The experimental results are depicted in Fig. 7. According to the figure, when R_b is larger than R_w and their gap is gradually increasing, the classification performance of the proposed method gets commensurately worse. This is caused by the aggravation of data imbalance in the established graph model. Another observation worth noting from the results on the ETH-80 dataset is the insensitivity of the proposed method to the parameters. For the MDSD dataset, our algorithm is also insensitive when R_w and R_b vary in the range of $\{5, 6, 7\}$ and $\{4, 5, 6\}$. Accordingly, we set R_w, R_b for the ETH-80 and MDSD datasets to 5, 3 and 5, 5, respectively. Their corresponding values on the AFEW and YTC datasets are set as 16, 16 and 3, 2, respectively (see Table I).

G. Ablation Study for the Convergence Behavior

As discussed in Section III-C, we expect to study the transformation matrix U but have to infer δ_q and h_q simultaneously. Therefore, we use an iterative method to solve this optimization problem, defined in Eq. (21). Although, it is difficult to provide a theoretical proof of convergence of this optimization process, our experience is that after several iterations the objective function Eq. (21) reaches a stable value. Fig. 8 was obtained using the AFEW and YTC datasets. It is evident that with increasing number of iterations, the value of objective function tends to decrease and eventually fluctuates within a very small range. This demonstrates the proposed model has a good convergence properties. The maximum number of iterations we set for the

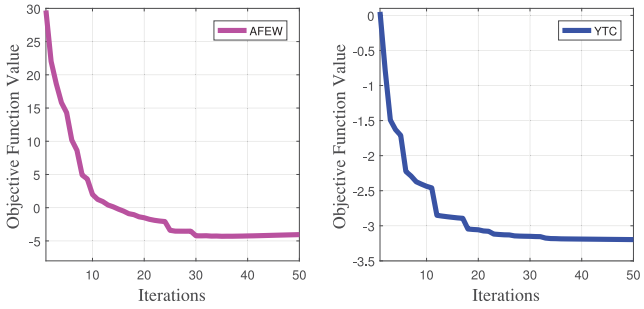


Fig. 8. Convergence behavior of the proposed algorithm on the AFEW and YTC datasets.

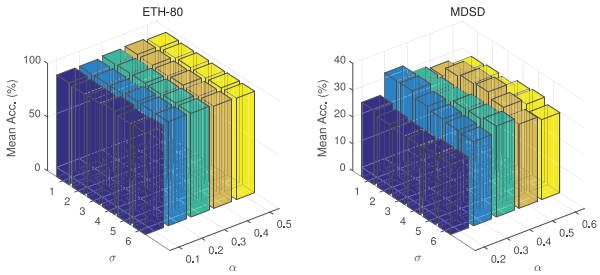


Fig. 9. Mean classification results of GEMKML on the ETH-80 and MDSD datasets under different parameter settings. α and σ take values from the sets $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $\{1, 2, 3, 4, 5, 6\}$ for ETH-80. For the MDSD dataset, they are take values from the sets $\{0.2, 0.3, 0.4, 0.5, 0.6\}$ and $\{1, 2, 3, 4, 5, 6\}$.

AFEW, YTC and ETH-80 datasets is 40, and for the MDSD 20 is sufficient.

H. Parameter Discussion

As described in Section III-B, the parameter α in the defined affinity matrix Λ is used to balance the intra-class compactness and the inter-class dispersion. σ impacts on the graph model which controls the weight assigned to each pair of image sets. To assess their influence, we experiment on the MDSD and ETH-80 datasets and measure the classification performance as a function of α and σ. From the results in Fig. 9, we note: 1) when σ is fixed, the classification performance of the proposed GEMKML tends to increase first and then drop with increasing α on both datasets. This suggest that α can play a part in balancing the within-class compactness and the between-class dispersion; 2) When changing both α and σ, the experimental results of our model tend to change slowly, suggesting a fair degree of robustness. The recommended values of the parameters that were used on the four used datasets are given in Table I.

Recalling the foregoing discussion, in the proposed GEMKML framework, the high dimensional Grassmann manifold-valued features are fused in a d_b-dimensional common subspace. Since more discriminative representations often reside in a lower dimensional feature space, it is indispensable to find an appropriate value of d_b. Taking the MDSD dataset as an example, we vary d_b and measure its impact on the classification rate in our experiments. The experimental results are shown in Fig. 10. It is interesting to see that the classification performance of the proposed method monotonically increases

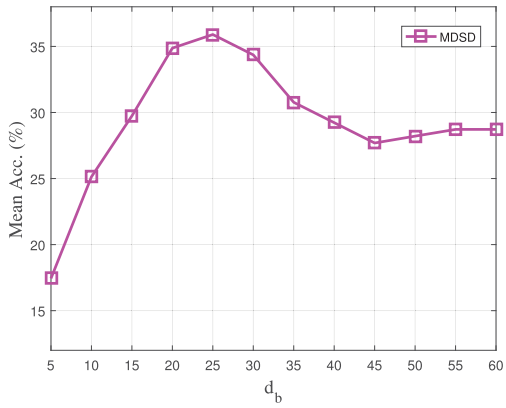


Fig. 10. Mean classification results of GEMKML on the MDSD dataset under different parameter settings, where d_b takes values from the range of $\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60\}$.

when d_b is in the range of 5$\sim$25. Once it exceeds 25, the classification rate of our algorithm tends to change smoothly. The top classification accuracy is achieved when $d_b = 25$. Increasing the dimensionality to 91, its maximum value for this dataset, the resulting classification rate of 30.26% is somewhat lower than its maximum.

The behaviour of the results in Fig. 10 can be explained as follows. When the value of d_b is low, the learned feature information is insufficient to differentiate between similar and dissimilar pairs, which compromises the classification ability. When the value of d_b is high, the learned representation includes redundant information, which also degrades the recognition performance. The same analysis on the AFEW, YTC and ETH-80 datasets suggests the values of d_b should be set set to 70, 70 and 10, respectively (see Table I).

I. Computational Time Comparison

To evaluate the time efficiency of the proposed method, we compare its average training and testing times with some representative image set classification methods on the YTC dataset. The experiments were run on Windows 10 Operating System with 3.0 GHz PC and 16 GB RAM. We need to emphasise that the CPU time is included in the running time. The testing time is defined by measuring the time required to classify one image set (one video sequence) into its category. The experimental results for all the methods obtained with the Matlab2016a software are shown in Table VI. From the table, we can see that the proposed cascaded feature extraction architecture (CasArct) exhibits a significantly reduced computational burden for both training and testing, compared to all the competitors, which proves the merits of using PCA and (2D)^{2}PCA to train the model. Note also that the testing time of the proposed GEMKML is higher than that of SPDNet and GrNet, mainly because GEMKML requires extra time to construct the m_1 Grassmannian manifold kernel matrices in the testing stage.

V. Application to 3D Hand Pose Estimation

While action recognition is a longstanding problem in the computer vision community, the first-person view based 3D

TABLE VI
RECOGNITION SCORES OF THE DIFFERENT METHODS ON THE FPHA DATASET

Method	Year	Color	Depth	Pose	Accuracy
JOULE-pose [53]	2015	✗	✗	✓	74.60
JOULE-all [53]	2015	✓	✓	✓	78.78
Two stream-color [46]	2016	✓	✗	✗	61.56
Two stream-flow [46]	2016	✓	✗	✗	69.91
Two stream-all [46]	2016	✓	✗	✗	75.30
Novel View [47]	2016	✗	✓	✗	69.21
HBRNN [48]	2015	✗	✗	✓	77.40
1-layer LSTM [49]	2016	✗	✗	✓	78.73
2-layer LSTM [49]	2016	✗	✗	✓	80.14
TCN-16 [50]	2017	✗	✗	✓	76.28
TCN-32 [50]	2017	✗	✗	✓	78.57
H+O [51]	2019	✓	✗	✗	**82.43**
TF [52]	2017	✗	✗	✓	80.69
PML [42]	2015	✗	✗	✓	75.48
GrNet [30]	2018	✗	✗	✓	77.57
CasArct		✗	✗	✓	74.50
GEMKML		✗	✗	✓	81.75

Fig. 11. Some 3D hand pose instances of the FPHA dataset.

hand pose estimation has made considerable advance since the recent availability of RGB-D sensors. The task of 3D hand action recognition is to estimate what kind of hand action is being performed by comprehending a video sequence via 3D coordinates of the joints. Because of the large occlusions and fast motion created by an action, the hand action recognition from first-person views pose a unique challenge. As each video sequence can be viewed as an image set, we investigate the applicability of the proposed approach to the FPHA dataset [45].

FPHA is a large and diverse first-person hand action dataset for 3D hand pose estimation. It consists of 1,175 hand action videos belonging to 45 different categories, performed by 6 actors in 3 different scenarios. In this dataset, a total of 105,459 RGB-D frames have been annotated with accurate hand poses and action categories, and a wide range of intra-subject and inter-subject variability of style, speed, scale, and viewpoint is presented in each video. Some 3D hand pose instances of the FPHA dataset are shown in Fig. 11. In this section, we follow the standard protocol of [45] to conduct the experiments. Specifically, we first normalize each video sequence to include 50 frames. Then, we characterize each hand gesture frame by transferring it into a 63-dimensional vector using the 3D coordinates of 21 hand joints provided. As a result, a feature matrix $S_i \in \mathbb{R}^{63\times 50}$ is obtained to represent a given hand pose video clip V_i. Lastly, 600 hand action video sequences are utilized to train the proposed model, and the remaining 575 action sequences are used for testing.

We compare the proposed approach with state-of-the-art deep learning-based action recognition methods, such as convolutional two-stream network (Two stream) [46], Novel View [47], hierarchical recurrent neural network (HBRNN) [48], LSTM [49], temporal convolution network (TCN) [50], and unified hand and object model (H+O) [51]. We also include transition forests (TF) [52], jointly learning heterogeneous features (JOULE) [53], PML [14], and GrNet [13] as the comparative algorithms. The recognition results of the different methods are listed in Table VII. Note that, we run the approaches of TCN, PML, and GrNet with their publicly available codes, and report their best recognition scores. The experimental results of H+O and other competitors are from [51] and [45], respectively. As to TCN, we run it with two different convolutional kernel lengths, 16 and 32, respectively. For the proposed GEMKML, the values of the fundamental parameters q, d_1, d_4, m_1, R_w, R_b, α, δ, and d_b, are set to 10, 31, 15, 2, 10, 10, 0.45, 0.5, and 40, respectively.

From Table VII we find: 1) The Grassmannian manifold-based learning approaches (PML and GrNet) exhibit comparable classification performance. This provides further demonstration of the effectiveness of Riemannian geometry in modeling the nonlinear interactions of different frames within the video. 2) The recognition score of the proposed cascaded feature extractor (CasArct) is 74.50%, lower than most of the competitors on this dataset. The reason stems from its inherent unsupervised learning mechanism and shallow architecture. 3) Although the recognition ability of GEMKML is inferior to the H+O method, our model is still competitive with most of the competitors on this dataset, with the advantage of much lower computational complexity. This validates the utility of the proposed image set classification approach.

VI. CONCLUSION

In this paper, we developed a novel algorithm for image set classification. Our contributions include a cascaded feature extraction architecture constructed on the Grassmannian manifold, and a new Grassmannian manifold-valued feature representation, facilitating the final classification. We also proposed a graph embedding multi-kernel metric learning framework to learn an adaptive distance metric for combining the extracted features. The results of extensive classification experiments on five datasets demonstrate the superiority of the proposed approach over representative image set classification methods.

For future work, one possible direction is to study and integrate Grassmannian deep learning networks into GEMKML for the purpose of learning deep Grassmannian feature representation. Another is to investigate other metric learning methods to further improve discriminatory power for more challenging scenarios. In addition, we plan to provide theoretical underpinning of the convergence properties of the proposed GEMKML.

REFERENCES

[1] W. Wang, R. P. Wang, Z. W. Huang, S. G. Shan, and X. L. Chen, "Discriminant analysis on Riemannian manifold of Gaussian distributions for face recognition with image sets," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 2048–2057.
[2] J. H. Hamm and D. D. Lee, "Grassmann discriminant analysis: A unifying view on subspace-based learning," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 376–383.
[3] Z. F. Wu, Y. Z. 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. P. Wang, H. M. Guo, L. S. Davis, and Q. H. Dai, "Covariance discriminative learning: A natural and efficient approach to image set classification," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2010, 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. Int. Conf. Pattern Recognit.*, 2018, pp. 627–632.

[6] 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*, 2020, doi: 10.1109/TBDATA.2020.2982146.

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

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

[9] M. T. 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] J. W. Lu, V. E. Liong, and J. Zhou, "Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1979–1993, Aug. 2018.

[11] Z. W. Huang, R. P. Wang, S. G. Shan, L. Van Gool, and X. L. Chen, "Cross euclidean-to-riemannian metric learning with application to face recognition from video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2827–2840, Dec. 2018.

[12] Y. M. Rao, J. W. Lu, and J. Zhou, "Learning discriminative aggregation network for video-based face recognition and person re-identification," *Int. J. Comput. Vision*, vol. 127, pp. 701–718, 2019.

[13] Z. W. Huang, J. Q. Wu, and G. L. Van, "Building deep networks on Grassmann manifolds," in *Proc. AAAI*, 2018, pp. 3279–3286.

[14] Z. W. Huang, R. P. Wang, S. G. Shan, and X. L. Chen, "Projection metric learning on Grassmann manifold with application to video based face recognition," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 140–149.

[15] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell, "Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2011, pp. 2705–2712.

[16] A. Wiliem, R. Vemulapalli, and B. C. Lovell, "Explicit discriminative representation for improved classification of manifold features," *Pattern Recognit. Lett.*, vol. 80, pp. 121–128, 2016.

[17] S. K. Chen, C. Sanderson, M. T. Harandi, and B. C. Lovell, "Improved image set classification via joint sparse approximated nearest subspaces," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2013, pp. 452–459.

[18] X. J. Wu, J. Kittler, J. Y. Yang, and S. T. Wang, "An analytical algorithm for determining the generalized optimal set of discriminant vectors," *Pattern Recognit.*, vol. 27, pp. 1949–1952, 2004.

[19] L. Zhang, B. P. Ma, G. R. Li, Q. M. Huang, and Q. Tian, "Cross-modal retrieval using multiordered discriminative structured subspace learning," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1220–1233, Jun. 2017.

[20] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, pp. 303–353, 1998.

[21] M. T. Harandi, C. Sanderson, C. H. Shen, and B. C. Lovell, "Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 3120–3127.

[22] J. W. 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. Vision*, 2013, pp. 329–336.

[23] Z. W. Huang, R. P. Wang, S. G. Shan, and X. L. Chen, "Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning," *Pattern Recognit.*, vol. 48, pp. 3113–3124, 2015.

[24] S. C. Yan *et al.*, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[25] H. Y. Cai, V. W. Zheng, and K. V. V. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.

[26] Gönen. M and E. Alpaydin, "Localized multiple kernel learning," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 352–359.

[27] P. A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[28] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell, "Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach," in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 216–229.

[29] Z. W. Huang and G. L. Van, "A Riemannian network for SPD matrix learning," in *Proc. AAAI*, 2017, pp. 2036–2042.

[30] J. W. Lu, G. Wang, W. H. Deng, P. Moulin, and J. Zhou, "Multi-manifold deep metric learning for image set classification," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1137–1145.

[31] X. S. Nguyen, L. Brun, O. Lézoray, and S. Bougleux, "A neural network based on SPD manifold learning for skeleton-based hand gesture recognition," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 12036–12045.

[32] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 209–216.

[33] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. Math. Phys.*, vol. 7, pp. 200–217, 1967.

[34] 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.*, 2014, pp. 461–466.

[35] H. L. Sun *et al.*, "Learning deep match kernels for image-set classification," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 3307–3316.

[36] N. Shroff, P. Turaga, and R. Chellappa, "Moving vistas: Exploiting motion for describing scenes," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 1911–1918.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[38] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.

[39] H. Li and X. J. Wu, "DenseFuse: A fusion approach to infrared and visible images," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2614–2623, May 2019.

[40] V. E. Liong, J. W. Lu, Y. P. Tan, and J. Zhou, "Deep coupled metric learning for cross-modal matching," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1234–1244, Jun. 2017.

[41] M. T. Harandi, M. Salzmann, and M. Baktashmotlagh, "Beyond Gauss: Image-set matching on the Riemannian manifold of pdfs," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 4112–4120.

[42] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, pp. 2385–2404, 2000.

[43] C. Ionescu, O. Vantzos, and C. Sminchisescu, "Matrix backpropagation for deep networks with structured layers," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 2965–2973.

[44] 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, pp. 328–347, 2007.

[45] 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. Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 409–419.

[46] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1933–1941.

[47] H. Rahmani and A. Mian, "3D action recognition from novel viewpoints," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1506–1515.

[48] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1110–1118.

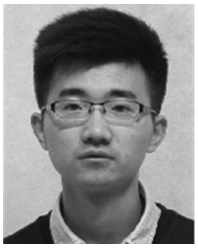
[49] W. Zhu *et al.*, "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks," in *Proc. AAAI*, 2016, pp. 3697–3703.

[50] T. S. Kim and A. Reiter, "Interpretable 3D human action analysis with temporal convolutional networks," in *Proc. Conf. Comput. Vision Pattern Recognit. Workshops*, 2017, pp. 1623–1631.

[51] B. Tekin, F. Bogo, and M. Pollefeys, "H+O: Unified egocentric recognition of 3D hand-object poses and interactions," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 4511–4520.

[52] G. Garcia-Hernando and T.-K. Kim, "Transition forests: Learning discriminative temporal transitions for action recognition and detection," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 432–440.

[53] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang, "Jointly learning heterogeneous features for RGB-D activity recognition," in *Proc. Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 5344–5352.



Rui Wang received the M.S. degree in the school of Internet of Things Engineering, Jiangnan University, Wuxi, China, in 2018. He is currently working toward his Ph.D. degree at the Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi, China. 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. degree and Ph.D. degree 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, 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. He 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, in 1971, 1974, and 1991, respectively. He is a distinguished Professor of Machine Intelligence at 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 61,000 times (Google Scholar). He is 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 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.