More About Covariance Descriptors for Image Set Coding: Log-Euclidean Framework based Kernel Matrix Representation

Kai-Xuan Chen[1] Xiao-Jun Wu[1,*] Jie-Yi Ren[1,2] Rui Wang[1] Josef Kittler[2]
[1]School of IoT Engineering, Jiangnan University,Wuxi, 214122, China.
[2]Center for Vision, Speech and Signal Processing, University of Surry, GU2 7XH, Guildford, UK.
kaixuan_chen_jnu@163.com wu_xiaojun@jiangnan.edu.cn jieyi.ren@surrey.ac.uk
cs_wr@jiangnan.edu.cn j.kittler@surrey.ac.uk

Abstract

We consider a family of structural descriptors for visual data, namely covariance descriptors (CovDs) that lie on a non-linear symmetric positive definite (SPD) manifold, a special type of Riemannian manifolds. We propose an improved version of CovDs for image set coding by extending the traditional CovDs from Euclidean space to the SPD manifold. Specifically, the manifold of SPD matrices is a complete inner product space with the operations of logarithmic multiplication and scalar logarithmic multiplication defined in the Log-Euclidean framework. In this framework, we characterise covariance structure in terms of the arc-cosine kernel which satisfies Mercer's condition and propose the operation of mean centralization on SPD matrices. Furthermore, we combine arc-cosine kernels of different orders using mixing parameters learnt by kernel alignment in a supervised manner. Our proposed framework provides a lower-dimensional and more discriminative data representation for the task of image set classification. The experimental results demonstrate its superior performance, measured in terms of recognition accuracy, as compared with the state-of-the-art methods.

1. Introduction

The representation of visual data plays a vital role in identifying the content of images [14,23], image sets [5,28] and videos [8,21]. There are many descriptors for visual data. As is well known, the most popular representations for recognition tasks are bag-of-visual-words (BoVW) models [25], fisher vectors (FV) [14] and vector of locally aggregated descriptors (VLAD) [1]. These representations are ultimately in the form of vectors and typically involve the following four steps: extracting features, generating codebook, encoding and pooling, and normalization.

Structured representations, such as linear subspaces [9], and covariance descriptors (CovDs) [29,32], have recently been shown to offer efficient and powerful representations for high dimensional tasks in computer vision. In particular, CovDs, defined by the second-order statistics of sample features, have been widely used as the visual representations for both single image [29] and image sets [32]. Prior to CovDs for describing image sets studied in [32], covariance matrices have been used as region covariance descriptors to characterise local regions within an image. They have been applied to the task of object tracking, object detection and texture classification. In contrast, when CovDs are used to describe image sets [32], samples are the images in the set and features are the raw intensities of the image pixels. The resulting covariance matrices are often singular because the feature dimensionality (the number of the pixels in the image) is usually larger than the number of samples (the number of images). Among the aforementioned methods, CovDs for describing image sets has the following characteristics:

1) In contrast to the vector representations of BoVW, FV and VLAD, which generate linear descriptors via codebooks, CovDs directly generate structured representations.

2) A feature matrix of an image set with n images: S = $[s_1,s_2,...,s_n]$, where $s_i \in R^d$ is a d-dimensional feature vector characterising the i-th image. Here, d is the number of the pixels in each image.

3) The resulting covariance matrix $C \in R^{d\times d}$ tends to be singular and high dimensionality, and may contain a certain amount of redundant information.

Characteristic 3 summarises several deficiencies of traditional CovDs as image set descriptors. In this paper, we propose a improved framework, which involves using a kernel matrix defined in terms of representations associated with sub-image sets, instead of pixels. Our proposed framework enables to generate covariance descriptors on non-linear SPD manifold (CovDs-S[1]). The experimental results

[1]Source code: https://github.com/Kai-Xuan/iCovDs

show the advantages of our proposed CovDs-S.

The rest of this paper is organized as follows: In Section 2, we introduce the theory of Riemannian geometry of SPD manifold and review the Log-Euclidean framework which is the baseline for our proposed approach. In Section 3, we give a brief overview of the traditional CovDs as image set descriptors and present the proposed framework. We present and discuss the experimental results in Section 4. Section 5 draws conclusions and outlines future work.

2. Background Theory

This section first provides a brief introduction to Riemannian geometry of SPD manifold, and proposes the process of SPD mean centralization. We then present a Log-Euclidean framework based arc-cosine (LogE.Arc) kernel.

Notation: In this paper, I_n is an $n \times n$ identity matrix. S_n^{++} denotes the SPD manifold spanned by real $n \times n$ SPD matrices and S_n denotes the space spanned by real $n \times n$ symmetric matrices. $T_P S_n^{++}$ is the tangent space at the point $P \in S_n^{++}$, which is a flat surface spanned by real $n \times n$ symmetric matrices. Diag$(e_1, e_2, ..., e_n)$ is a diagonal matrix with the diagonal elements $e_1, e_2, ..., e_n$. The matrix logarithm, log$(\cdot)$: $S_n^{++} \to S_n$ is defined as:

$$\log(X) = U\text{Diag}((\log(e_1, e_2, ..., e_n))U^T \tag{1}$$

with $X = U\text{Diag}(e_1, e_2, ..., e_n)U^T$. If $X \in S_n^{++}$ is an SPD matrix, log$(X) \in T_I S_n^{++}$ will be a point in the tangent space at the identity matrix I_n. Similarly, the matrix exponential exp$(\cdot)$: $S_n \to S_n^{++}$ is defined as:

$$\exp(X) = U\text{Diag}((\exp(e_1, e_2, ..., e_n))U^T \tag{2}$$

with $X = U\text{Diag}(e_1, e_2, ..., e_n)U^T$.

2.1. The General Metrices on SPD Manifold

A real $n \times n$ SPD matrix $X \in S_n^{++}$ satisfies $v^T X v \geq 0$ for all non-zero $v \in R^n$. The Affine Invariant Riemannian Metric (AIRM) is the frequently studied Riemannian metric on the SPD manifold [22]. Beside AIRM, Log-Euclidean Metric (LEM) [2] and two types of Bregman divergence [16], namely Stein [26] and Jeffrey [11] divergence, are also widely used to analyze SPD matrices.

Definition 1 (Affine Invariant Riemannian Metric, AIRM) *The most common Riemannian metric on S_n^{++} is Affine Invariant Riemannian Metric (AIRM) [22], in which the geodesic distance d_G: $S_n^{++} \times S_n^{++} \to [0, \infty)$ between two SPD matrices X and Y can be obtained by:*

$$d_G^2(X, Y) = \| \log \left(X^{-\frac{1}{2}} Y X^{-\frac{1}{2}} \right) \|_F^2 \tag{3}$$

where $\| \cdot \|_F$ denotes is the Frobenius norm. log$(\cdot)$ is the matrix principal logarithm.

Definition 2 (Stein divergence) *The Stein, or S, divergence [26] d_S: $S_n^{++} \times S_n^{++} \to [0, \infty)$ is a special type of Bregman divergence:*

$$d_S^2(X, Y) = \log\det(\frac{X+Y}{2}) - \frac{1}{2}\log\det(XY) \tag{4}$$

Definition 3 (Jeffrey divergence) *The Jeffrey, or, J, divergence [11] d_J: $S_n^{++} \times S_n^{++} \to [0, \infty)$ is another type of Bregman divergence:*

$$d_J^2(X, Y) = \frac{1}{2}Tr(X^{-1}Y) + \frac{1}{2}Tr(Y^{-1}X) - n \tag{5}$$

The Stein and Jeffrey divergence are similar to geodesic distance induced by AIRM [22].

Definition 4 (Log-Euclidean Metric, LEM) *For two SPD matrices $X, Y \in S_n^{++}$, the Log-Euclidean Distance (LED) [2, 32], is defined by Frobenius norm in the tangent space at identity matrix I_n:*

$$d_{LEM}^2(X, Y) = \| \log(X) - \log(Y) \|_F^2 \tag{6}$$

Accordingly, the SPD manifold will be reduced to a flat Riemannian space [2] while endowed with LEM.

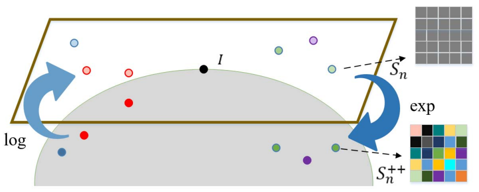

Figure 1. The illustration of logarithmic and exponential mapping.

2.2. Log-Euclidean Framework

In the Log-Euclidean framework, the matrix logarithm log$(\cdot)$: $S_n^{++} \to S_n$ is smooth and bijective, and its inverse map, denoted by exp$(\cdot)$, is smooth as well. Figure 1 illustrates these two operations on SPD manifold. The Log-Euclidean Kernel [32] is derived by computing the inner product in the domain of logarithm matrix:

$$k_{LogE}(X, Y) = Tr(\log(X)\log(Y)) \tag{7}$$

where Tr denotes the matrix trace. The Log-Euclidean kernel is a positive definite kernel and has been shown to meet Mercer's conditions in [32]. The operations logarithmic multiplication and scalar logarithmic multiplication are the corresponding Euclidean operations in the domain of logarithm matrix, followed by an inverse mapping back to the SPD manifold via the operation of matrix exponential (Interested readers can refer to [2, 19] for details). We thus can propose the operation of mean centralization on SPD matrices.

Proposition 1 *In line with the brief overview of the Log-Euclidean framework [2, 19], we define the operation of mean centralization on SPD matrices in three steps. Firstly, we map the SPD matrices into the domain of logarithm matrix. Then, we centralize the resulting symmetric matrix by an operation that is similar to centering the kernel matrix in [7]. Finally, we map the centralized matrices back to SPD manifold via exponential mapping. For an arbitrary real $n \times n$ SPD matrix $X \in S_n^{++}$, the operation of mean centralization can be written as:*

$$\begin{aligned}\tilde{X} &= \exp(\hat{X}) \\ where \quad [\hat{X}]_{i,j} &= [\log(X)]_{i,j} - \frac{1}{n}\sum_{i=1}^{n}[\log(X)]_{i,j} \\ &- \frac{1}{n}\sum_{j=1}^{n}[\log(X)]_{i,j} + \frac{1}{n^2}\sum_{i,j=1}^{n}[\log(X)]_{i,j}\end{aligned} \tag{8}$$

Here, $\tilde{X}$ is the result of our proposed mean centralization operation applied to the SPD matrix X.

Inspired by the broad applications of arc-cosine kernel [6] in the Euclidean space and a family of Log-Euclidean kernels proposed in [19], we propose Log-Euclidean framework based arc-cosine kernel(LogE.Arc kernel), which extends the well-known arc-cosine kernel onto the nonlinear Riemannian manifold of SPD matrices.

Definition 5 (arc-cosine kernel) *Let $x, y \in R^d$ be two vectors. The arc-cosine kernel can be expressed as the angle θ between the samples [6] as:*

$$\theta = \arccos\left(\frac{x \cdot y}{\| x \| \| y \|}\right) \tag{9}$$

In [6], the arc-cosine kernel has a simple formulation, which depends on the magnitude of the vectors and the angle between them. It can be defined as:

$$k_r(x, y) = \frac{1}{\pi}\| x \|^r \| y \|^r J_r(\theta) \tag{10}$$

where the angular dependence function $J_r(\theta)$ for different orders $r \geq 0$ is defined as:

$$J_r(\theta) = (-1)^r (\sin\theta)^{2r+1}\left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\right)^r\left(\frac{\pi - \theta}{\sin\theta}\right) \tag{11}$$

The arc-cosine kernel function $k_r(x, y)$ has different properties that are shared respectively by radial basis function (RBF), linear, and polynomial kernels (Interested reader can refer to [6] for the details of the arc-cosine kernel). Motivated by the work in [19] and the Log-Euclidean framework, the inputs, x, and y, of arc-cosine kernel can not only be vectors in the Euclidean space, but also SPD matrices on the curved Riemannian manifold. Thus, the Log-Euclidean framework based arc-cosine kernels can be defined as:

$$k^r_{LogE.Arc}(x, y) = \frac{1}{\pi}\| \log(x) \|_F^r \| \log(y) \|_F^r J_r(\theta) \tag{12}$$

Here $x, y \in S_n^{++}$ and $J_r(\theta)$ has the same formulation as Eq.11, θ is the angle between the inputs that are mapped into the domain of logarithm matrix:

$$\theta = \arccos\left(\frac{Tr(\log(x)\log(y))}{\| \log(x) \|_F \| \log(y) \|_F}\right) \tag{13}$$

The arc-cosine kernel given in (Eq.12) sets up a Log-Euclidean framework for constructing kernels on SPD manifold, which measures the similarity of SPD matrices and referred to as Log-Euclidean framework based arc-cosine kernel (LogE.Arc kernel). The LogE.Arc kernels of different orders are the corresponding arc-cosine kernels in the domain of logarithm matrix which inherit the corresponding property (RBF, etc.) in the vector space.

3. Proposed Framework

In this section, we first give a brief overview of traditional CovDs for describing image sets and then present our proposed CovDs-S. Finally, we compare our CovDs-S with other improved versions of CovDs.

3.1. Traditional Covariance Descriptors

Consider a feature matrix (Fig.2 step (b)) of an image set with n images: $S = [s_1, s_2, ..., s_n]$, where $s_i \in R^d$ is the d-dimensional feature vector representing the i-th image. Color images need to be processed as grayscale images and the d-dimensional feature vectors are obtained by vectorizing the grayscale images. Using the traditional CovDs, the representation [32] of this image set can be obtained by:

$$\begin{aligned}C &= \frac{1}{n-1}\sum_{i=1}^{n}(s_i - \tilde{s})(s_i - \tilde{s})^T = \tilde{S}\tilde{S}^T \\ where \quad \tilde{S} &= (n-1)^{-\frac{1}{2}}(S - \tilde{s}1_n^T)\end{aligned} \tag{14}$$

and $\tilde{s} = \frac{1}{n}\sum_{i=1}^{n} s_i$ is a d-dimensional mean vector of the feature matrix S. $\tilde{S}$ is the mean centralized matrix (Fig.2 step (c)), and 1_n is a column vector of n ones. The covariance matrix, C, can also be viewed as the kernel matrix between mean centralized feature vectors of the corresponding pixels (the rows of mean centralized matrix $\tilde{S}$) via linear kernel (Fig.2 step (d)):

$$\begin{aligned}C &= (C_{i,j})_{i,j=1,...,d} \\ where \quad C_{i,j} &= k_{linear}(\tilde{S}_{(i,:)}, \tilde{S}_{(j,:)}) = \tilde{S}_{(i,:)}\tilde{S}^T_{(j,:)}\end{aligned} \tag{15}$$

where $\tilde{S}_{(i,:)} \in R^n$ denotes the i-th row of $\tilde{S}$, which is also the feature vector that represents i-th pixel of n images. $C_{i,j}$ is the result of a linear kernel operation between $\tilde{S}_{(i,:)}$ and $\tilde{S}_{(j,:)}$ and denotes the similarity between i-th pixel and j-th pixel.

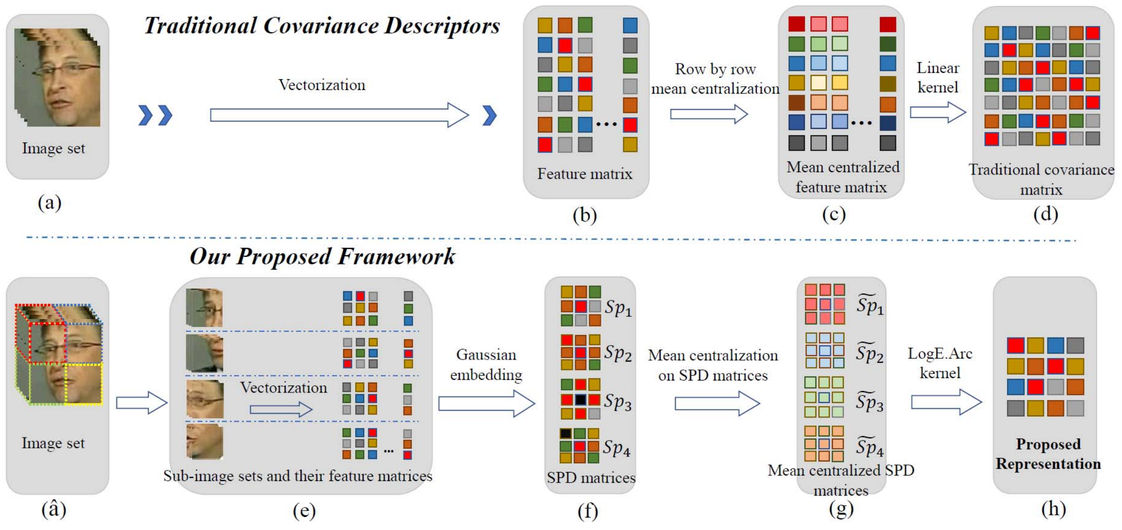

Figure 2. The flow chart of traditional covariance descriptors (CovDs) and our proposed framework (CovDs-S). Top: traditional CovDs for describing image set. Bottom: our proposed framework for describing image set.

3.2. Proposed Framework for Image Set Coding

Our proposed framework offers lower-dimensional and more discriminative representation for describing image sets than the traditional CovDs. For the sub-volumes of an image set (Fig.2 step (â)), namely sub-image sets, we use SPD matrices to describe them via a Gaussian embedding. We then centralize these SPD matrices and use the LogE.Arc kernel to operate on the resulting mean centralized SPD matrices. Finally, the image set representation in our proposed framework is the kernel matrix defined by the mean centralized SPD matrices associated with the corresponding sub-image sets. We first give a brief overview of the Gaussian embedding and elaborate the bottom row of Figure 2 to describe our framework.

The feature matrix $S = [s_1, s_2, ..., s_n]$, as introduced in the sub-section of traditional CovDs, can also be described by a Gaussian model. The space spanned by a Gaussian model is a Riemannian manifold, and Gaussian embedding can embed this special manifold into SPD manifold [31]:

$$N(\mu, \Sigma) \sim G(\beta) = \begin{bmatrix} \Sigma + \beta^2 \mu\mu^T & \beta\mu \\ \beta\mu^T & 1 \end{bmatrix} \tag{16}$$

where μ is a d-dimensional mean vector of S and Σ is a real $d \times d$ covariance matrix. $\beta > 0$ is a parameter balancing the covariance matrix and the mean vector, The resulting matrix $G(\beta) \in R^{(d+1)\times(d+1)}$ is an SPD matrix and used as descriptors for sub-image sets(Fig.2 step (f)) in our proposed framework.

The bottom row of Figure 2 shows the flow chart of our proposed framework. We first obtain 4 sub-image sets via a sliding window(Fig.2 step (â), *where the particular 4 sub-image were selected just for demonstration. Their choice is not fixed in our framework*). Then we use four SPD matrices (Fig.2 step (f)) to represent sub-image sets via Gaussian embedding (Eq.8) and obtain four mean centralized SPD matrices: $\tilde{Sp}_1$, $\tilde{Sp}_2$, $\tilde{Sp}_3$, $\tilde{Sp}_4$(Fig.2 step (g)) via the operation of mean centralization. Finally, the resulting representation (Fig.2 step (h)) is the sum kernel matrix of the four mean centralized SPD matrices via the LogE.Arc kernels of different orders(Eq.12). To this end, the resulting representation can generally be defined as:

$$\begin{aligned} C_{CovDs-S} &= \sum_{r=0}^{R} w_r C_r \\ where \quad C_r &= ([C_r]_{i,j})_{i,j=1,...,N} \\ and \quad [C_r]_{i,j} &= k^r_{LogE.Arc}(\tilde{Sp}_i, \tilde{Sp}_j) \end{aligned} \tag{17}$$

where N is the number of the sub-image sets, which is the key parameter determining the dimensionality, R is the number of orders selected for LogE.Arc kernel. $\tilde{Sp}_i$ is the mean centralized SPD matrix, which is the representation of the i-th sub-image set. C_r is the local kernel matrix using the r-th order LogE.Arc kernel function between mean centralized SPD matrices: $\tilde{Sp}_1$,$\tilde{Sp}_2$,...,$\tilde{Sp}_N$. The resulting representation: $C_{CovDs-S}$ is the sum kernel matrix (The sum of kernels is also a $p.d$ kernel [27]) of the local kernel matrices multiplied by the corresponding weight w_r. $[C_{CovDs-S}]_{i,j}$ denotes the similarity of the i-th sub-image set and j-th sub-image set.

3.3. Learning weights via Kernel Alignment

In this sub-section, we learn, via the kernel target alignment, the weight coefficients w_r associated with the r-th order LogE.Arc kernel for our proposed framework via the kernel target alignment.

Definition 6 (Kernel Alignment [7,27]) *The kernel alignment aims to align an input kernel matrix K to a target kernel matrix K_T. It is defined as:*

$$\rho(K, K_T) = \frac{< K, K_T >_F}{\sqrt{< K, K >_F < K_T, K_T >_F}} \tag{18}$$

the result of Eq.18 can be viewed as the cosine of the angle between K and K_T. The weight coefficients $W = [w_0; ...; w_R]$ should be estimated by maximizing the $\rho(K_W, K_T)$. K_W is the global kernel matrix obtained as the sum of local kernel matrices of different orders. The kernel alignment has the following optimization formulation [7]:

$$W^* = \arg\max \rho(K_W, K_T) = \arg\max \frac{Tr(K_W K_T)}{\sqrt{Tr(K_W K_W)}} \tag{19}$$

We now introduce the kernel matrices: K_W and K_T. Given a set of training samples $X = [x_1, x_2, ..., x_N]$, where $x_i = \sum_{r=0}^{R} w_r C_r^i$ is our proposed representation for the i-th image set and C_r^i is the r-th order LogE.Arc kernel matrix between sub-image sets constructed from the i-th image set by image division, and the corresponding label matrix $Y = [y_1, y_2, ..., y_N]^T$ for the N samples, where $y_i \in R^C$ contains the class label information of i-th sample and the c-th element of y_i is 1 if x_i is from the c-th class. In this paper, the global kernel matrix K_W is the sum of K_w^r multiplied by the weight w_r: $K_W = \sum_{r=0}^{R} w_r K_w^r$ and K_w^r is the local kernel matrix between $C_r^1, ..., C_r^N$: $[K_w^r]_{i,j} = Tr(C_r^i C_r^j)$. The target kernel matrix K_T is defined via label matrix: $K_T = YY^T$. As introduced in [7], the objective function in Eq.19 can be rewritten as:

$$W^* = \underset{\|W\|=1}{\arg\max} \frac{W^T \beta \beta^T W}{W^T \Omega W} \tag{20}$$

where $\| W \| = 1$ is a regularization term. β is defined by $\beta_i = Tr(\hat{K}_w^i K_T)$, where $\hat{K}_w^i$ is the centralized matrix of the kernel matrix K_w^i [7,27], and the matrix Ω is defined by $\Omega_{i,j} = Tr(\hat{K}_w^i \hat{K}_w^j)$.

According to Proposition 2 in [7], the solution W^* of Eq.20 is given by:

$$W^* = \frac{\Omega^{-1}\beta}{\| \Omega^{-1}\beta \|} \tag{21}$$

3.4. Comparison with other Improved Versions of traditional CovDs for Image Set Coding

As far as we know, the works in [4, 20, 30] are the only three improved versions of traditional CovDs for describing image sets. It is desirable to manifest their connections and differences.

Wang et al [30] proposed an open framework[2] to use the kernel matrix over feature dimensions as a generic representation. This work uses a non-linear kernel matrix as the representation, but the kernel functions are defined in the Euclidean space and the resulting representation describes similarities between pixels at different locations, as the traditional CovDs [30]. Our work proposes to capture the similarities between sub-image sets that contain more useful information. It extends the acr-cosine kernel onto the SPD manifold for this purpose.

Li et al [20] extended the descriptive granularity of covariance matrix from traditional pixel-level to more general patch-level. Though this work concentrates on the patch-level covariance computation, it is actually a sum-pooling form of the pixel-level covariance. There is an essential difference in the way of descriptor computation. For describing image sets, we use the kernel matrix computed on SPD manifold as the resulting representation instead of the covariance matrix computed in the Euclidean space [20]. Moreover, we use the SPD matrices to represent sub-image sets instead of the feature matrices consisting of intensity values [20].

Chen et al [4] proposed a framework[3] to generate low-dimensional discriminative data representation for describing image sets and concentrated on characterising the similarities between sub-image sets instead of pixels. The main difference his approach and our method is that we combine arc-cosine kernels of different orders in the domain of logarithm matrix instead of using a linear kernel [4]. Moreover, we obtain our sub-image sets via the sliding window technique instead of dividing images into non-overlapping blocks [4] and use the Gaussian embedding to model them. The work in [4] is a special case of our proposed framework.

4. Experiments and Analysis

This section presents comparative experimental results of our proposed framework with state-of-the-art (SOTA) methods for the task of image set classification.

4.1. Datasets and settings

In our first experiment involving the task of image set classification, we consider the Cambridge hand-gesture (CG) dataset [15] that contains nine categories of samples and nine hundred image sets. Each class has twenty image sets chosen for training at random, and the remaining eighty image sets are reserved for testing. In the ETH-80 dataset [18], there are eight categories of samples and eighty image sets. For each class, five image sets are randomly chosen as training samples and the remaining five image sets are used for testing. In the Virus cell dataset [17], there

[2]https://www.uow.edu.au/ leiw/

[3]https://github.com/Kai-Xuan/ComponentSPD/

Table 1. Average recognition rates and standard deviations of different descriptors.

Methods	Descriptors	CG [15]	ETH-80 [18]	Virus [17]	MDSD [24]
NN-AIRM	CovDs [32]	51.82±2.55	70.12±5.24	27.57±4.34	13.08±4.05
	CovDs-B [30]	71.87±2.47	89.17±3.48	36.57±5.00	23.67±5.55
	CovDs-P [20]	87.49±1.41	87.83±4.61	67.40±6.19	22.51±4.56
	CovDs-C [4]	89.07±1.15	**94.53±2.55**	40.17±6.07	21.13±5.68
	CovDs-S (Ours)	**90.21±1.27**	94.18±3.69	**67.60±4.76**	**32.38±5.60**
NN-Stein	CovDs [32]	40.66±2.62	57.08±5.62	27.43±4.90	12.67±4.25
	CovDs-B [30]	75.06±2.32	88.32±3.71	35.17±4.48	22.92±6.17
	CovDs-P [20]	79.97±2.30	88.75±4.69	67.03±6.14	21.77±4.16
	CovDs-C [4]	89.76±1.23	94.10±2.90	39.83±6.18	19.51±5.42
	CovDs-S (Ours)	**90.30±1.28**	**94.18±3.69**	**67.70±4.80**	**31.51±5.89**
NN-Jeffrey	CovDs [32]	82.45±1.38	86.12±4.81	30.80±5.49	18.56±4.77
	CovDs-B [30]	59.38±2.42	90.13±3.58	40.03±5.81	23.67±5.37
	CovDs-P [20]	83.07±1.44	87.63±4.54	65.37±6.55	21.67±4.29
	CovDs-C [4]	89.52±1.14	94.60±2.79	40.97±6.54	24.00±5.30
	CovDs-S (Ours)	**90.02±1.22**	**94.68±3.64**	**67.50±4.79**	**33.03±5.72**
NN-LEM	CovDs [32]	67.47±1.93	78.17±5.59	25.97±4.62	13.74±4.52
	CovDs-B [30]	73.18±2.39	90.65±3.79	36.07±4.22	24.87±5.51
	CovDs-P [20]	89.44±1.17	88.27±4.24	67.57±6.61	24.90±4.96
	CovDs-C [4]	90.24±1.09	93.48±3.16	40.57±5.53	21.54±5.31
	CovDs-S (Ours)	**90.49±1.31**	**94.07±3.77**	**68.10±4.88**	**32.23±6.08**
Ker-SVM	CovDs [32]	91.54±1.16	92.60±5.15	65.83±5.63	35.74±6.11
	CovDs-B [30]	92.31±1.13	94.18±3.69	73.77±5.82	37.79±5.76
	CovDs-P [20]	94.34±1.02	94.52±3.64	75.40±6.01	35.54±6.47
	CovDs-C [4]	93.81±1.01	95.45±2.85	53.63±6.80	38.08±6.05
	CovDs-S (Ours)	**94.36±0.98**	**97.07±2.66**	**77.93±5.03**	**43.92±6.09**

are fifteen categories of samples and 100 images in each category. We divided the images of each category equally into five different image sets and obtained seventy-five image sets. For each class, three image sets are randomly chosen as training samples and the remaining two image sets for testing. The MDSD dataset [24] has been used for the task of dynamic scene classification. Following the settings in [27], we test the method based on the protocol of seventy-thirty-ratio (STR) which chooses seven videos for training and three videos for testing in each class.

4.2. A Comparison with Existing Descriptors

For the comparative experiments with existing descriptors [4,20,30], we first resize all images to 24×24 and then use the intensity values to generate their corresponding representations. For our proposed framework, the sub-image sets are obtained by 6×6 sliding window with spatial step of 2 pixels for the CG, ETH-80 and MDSD datasets, and spatial step of 3 pixels for the Virus dataset. In total, we obtain 100 sub-image sets for theCG, ETH-80 and MDSD datasets and 49 sub-image sets for the Virus dataset. In our framework, we set parameter r in LogE.Arc kernel to be r = [0, 1, 2, 3], and the value of β in Eq.16 depends on the datasets, which is 0.05, 0.9, 14, 2 for the four datasets respectively. For the learned $W = [w_0, ..., w_R]$, we set first two largest absolute values to be 1 and another two values to be zero on ETH-80, Virus and MDSD datasets. We set the largest absolute value to be 1 and another three values to be zero on theCG datasets. We regularize the traditional CovDs: $C^* = C + \lambda I$ to avoid the matrix singularity as introduced in [32], and set λ to $10^{-3} \times Tr(C)$. To generate the descriptor in [30], we obtain the final kernel matrix representation via RBF kernel that has been shown to produce better accuracies [30]. For the fairness of the comparative experiments, the patch size in [20] and block size in [20] are all 6×6 and the step size is the same as the setting used by our proposed CovDs-S on the corresponding dataset. The different descriptors evaluated in our experiments are referred to as:

CovDs: Image set repesented by traditional CovDs [32].
CovDs-B: Image set repesented by the method in [30].
CovDs-P: Image set repesented by the method in [20].
CovDs-C: Image set repesented by the method in [4].
CovDs-S: Image set repesented by our framework.

In our experiments, five classification algorithms are used to verify the validity of our proposed CovDs-S, which include four nearest neighbor (NN) algorithms based on AIRM, Stein divergence, Jeffrey divergence, LEM and the well-known SVM classifier [3]. The different methods tested in our experiments are referred to as:

NN-AIRM: AIRM-based NN classifier.
NN-Stein: Stein divergence-based NN classifier.

Table 2. Average recognition rates and standard deviations of different classifiers.

Methods	CG [15]	ETH-80 [18]	Virus [17]	MDSD [24]
COV-LDA [32]	90.25±1.64	93.95±4.30	46.40±5.76	34.10±5.90
COV-PLS [32]	88.95±1.26	94.23±4.63	62.84±5.99	36.74±5.62
LogEKSR.Pol [19]	92.32±1.19	95.00±3.28	58.53±6.54	36.23±6.81
LogEKSR.Exp [19]	92.23±1.18	95.10±3.20	59.03±6.38	36.59±6.88
LogEKSR.Gau [19]	92.33±1.18	95.18±3.30	61.80±6.35	37.95±6.83
LEML [13]	88.18±1.29	93.05±3.31	33.00±5.70	25.97±6.79
LEML+COV-LDA [13]	89.09±1.63	95.35±3.50	58.03±5.84	31.92±6.44
LEML+COV-PLS [13]	86.36±1.35	95.83±3.04	59.40±6.22	35.90±6.98
SPDML-LEM [10]	84.03±1.04	90.63±4.19	49.37±7.46	24.23±4.47
SPDNet [12]	92.03±1.46	95.50±3.69	59.70±4.58	33.76±5.04
MMML [33]	92.87±1.39	95.28±3.80	51.13±7.60	31.95±6.26
KS-CS-LEK	93.63±1.08	95.38±2.92	74.93±5.92	39.33±7.00
KS-CS-LogE.Pol	93.95±0.94	**97.30±2.55**	75.17±4.86	42.72±6.20
KS-CS-LogE.Exp	93.68±0.90	95.40±3.03	70.90±5.60	42.67±7.04
KS-CS-LogE.Gau	93.90±0.91	95.65±2.86	71.87±5.18	41.15±6.33
KS-CS-LogE.Arc	**94.36±0.98**	97.07±2.66	**77.93±5.03**	**43.92±6.09**

NN-Jeffrey: Jeffrey divergence-based NN classifier.
NN-LEM: LEM-based NN classifier.
Ker-SVM: LEK-based SVM classifier.

Here, Ker-SVM is a one-vs-all SVM classifier[4] implemented by Wang et al. Table 1 shows the average recognition rates and standard deviations of different descriptors with the same classifiers. In addition to the results with NN-AIRM on the ETH-80 dataset, the recognition rates of CovDs-S are higher than other four descriptors while using the same classification algorithm. This confirms that our CovDs-S captures more discriminative information than the other methods. In particular, the accuracy of CovDs-C is not as high as shown in Table 1 if the setting follows the recommendations made in [4].

4.3. Comparison with Existing Classifiers

Here, we compare our method with the SOTA algorithms including Covariance Discriminative Learning (COV-LDA, COV-PLS) [32], Log-Euclidean Kernels for Sparse Representation (LogEKSR.Pol, LogEKSR.Exp and LogEKSR.Gau) [19], SPD Manifold Learning based on LEM (SPDML-LEM) [10], Log-Euclidean Metric Learning (LEML, LEML+COV-LDA, LEML+COV-PLS) [13], Riemannian Network for SPD Matrix Learning (SPDNet) [12] and Multiple Manifolds Metric Learning (MMML) [33]. For these methods, we first resize all images to 20×20 and use the intensity values as their features.

We use the Ker-SVM tested on the representations obtained by our framework as our proposed classification algorithm. In addition to the LogE.Arc kernel (introduced above) used in our framework, we also consider other types of kernel functions to enrich our framework, such as Log-Euclidean Kernel (LEK) [32], Log-Euclidean based polynomial (LogE.Pol), exponential (LogE.Exp) and Gaussian kernels (LogE.Gau). Our framework based classifiers are reffered as:

KS-CS-LogE.Arc: Ker-SVM tested on CovDs-S (introduced above).
KS-CS-LEK: Ker-SVM tested on the representations via our framework where the LogE.Arc kernel replaced by LEK
KS-CS-LogE.Pol: Ker-SVM tested on the representations via our framework where the LogE.Arc kernelis replaced by LogE.Pol kernel
KS-CS-LogE.Exp: Ker-SVM tested on the representations obtained by our framework, where the LogE.Arc kernel is replaced by LogE.Exp kernel
KS-CS-LogE.Gau: Ker-SVM tested on the representations produced by our framework, where the LogE.Arc kernel is replaced by LogE.Gau kernel

As shown in Table 2, the classifiers based on our framwork produce better performance than other SOTA methods. The advantage of our methods is very obvious on the Virus and MDSD datasets, where image samples contain a large amount of noise. Our method KS-CS-LogE.Arc achieves the best recognition rate of 94.36%, 77.93% and 43.92% on the CG, Virus and MDSD datasets. Our framework based KS-CS-LogE.Pol achieves the best recognition rate of 97.30% on the ETH-80 dataset and KS-CS-LogE.Arc achieves the second best recognition rate of 97.07%.

4.4. Ablation Experiments

In this subsection, we validate the contributions of each component in CovDs-S and analyze the effect of image

[4]http://www.peihuali.org/publications/RAID-G/RIAD-G_V1.zip

Table 3. The accuracies and standard deviations of variants.

Descriptors	CG [15]	ETH-80 [18]	Virus [17]	MDSD [24]
CovDs-S-GE	94.18±0.88	96.73±2.90	60.00±6.21	40.10±6.68
CovDs-S-KA	91.76±1.26	96.62±3.02	74.63±5.14	43.79±6.16
CovDs-S-MC	94.34±0.97	97.07±2.66	77.67±4.68	43.74±6.10
CovDs-S-IM90	94.32±1.00	97.07±2.66	77.40±4.70	43.90±6.16
CovDs-S-IM180	94.34±0.97	97.07±2.66	77.83±4.95	43.92±6.06
CovDs-S-IM270	94.33±0.99	97.07±2.66	77.53±4.56	43.95±6.26

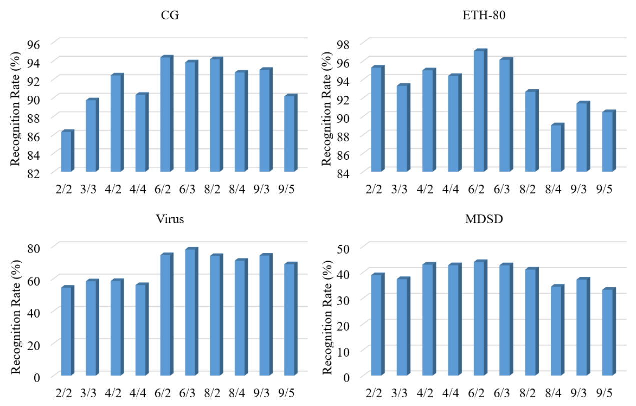

Figure 3. The infulence of different sub-image size and step size.

rotation and sub-image sizes. To this end, CovDs-S is used in the following variants: CovDs-S without Gaussian embedding (CovDs-S-GE), CovDs-S without kernel alignment (Coves-S-KA), CovDs-S without mean centralization (CovDs-S-MC), CovDs-S with image rotation 90° (CovDs-S-IM90), CovDs-S with image rotation 180° (CovDs-S-IM180) and CovDs-S with image rotation 270° (CovDs-S-IM270).

Table 3 shows the accuracies and the standard deviations of the different kernel variants in conjunction with the Ker-SVM classifier on the four datasets. From the results in this table, we can conclude that the contribution of Gaussian embedding and kernel alignment are more significant than the effect of mean centralization of our CovDs-S. In theory, the mean centralization operation corresponds to the standard normalization operation used in traditional CovDs. It appears that in the case of our framework, this operation does not impact on accuracy. According to the last three rows of the table, CovDs-S is also not sensitive to image rotation.

Figure 3 shows the effect of sub-image size and step size on the recognition rates of KS-CS-LogE.Arc. In the horizontal ordinate in the form of 'a/b', 'a' represents the sliding window size, and 'b' denotes step size. The results show clearly that the proposed framework performs bettter when the sub-image size is 6×6. In that case, our CovDs-S achieves the best accuracies on the CG, ETH-80 and MDSD datasets when the step size is 2 pixels. When the step size is set as 3 pixels, our CovDs-S achieves the best accuracies on the Viurs dataset.

Table 4. Time comparisons on ETH-80 dataset.

	CovDs	CovDs-S				
		LEK	LogE.Pol	LogE.Exp	LogE.Gau	LogE.Arc
GT	20.15	48.67	48.74	49.32	49.20	51.83
NN-AIRM	28.77	1.248	1.252	1.248	1.250	1.249
Ker-SVM	2.07	0.056	0.056	0.056	0.056	0.055

4.5. Advantages of Our Framework

From the superior results in Table 1 and Table 2, we can conclude that our proposed framework captures more discriminative information for the task of image set classification than other methods. The superior performance is particularly notable for noisy samples. Moreover, the dimensionality of the representation obtained via our framework is related to the number of sub-image sets(100 or 49), which is far lower than that of the traditional CovDs. Table 4 shows the run time for representative methods (NN-AIRM and Ker-SVM), as well as the generating time (GT) required for different descriptors on the ETH-80 dataset, where the unit of time is one second. The representations extracted by our framework require far less time than the traditional CovDs. With the recommended settings in the experiments, our proposed representations tend to be nonsingular. The dimensionality of the feature representation for the sub-image set is $(1 + (6^2 + 1))(6^2 + 1)/2 = 703$. The resulting representation can also be viewed as corresponding to the kernel matrix (arc-cosine kernel, etc. in the Euclidean space) of 100×703 or 49×703 feature matrix S as traditional CovDs. Here, we cannot be sure about the nonsingularity for the resulting kernel matrix, but it is more likely to hold for an SPD matrix than a traditonal covariance matrix.

5. Conclusion And Future Work

We proposed a novel framework extending CovDs from the Euclidean to an SPD manifold. It generates a kernel matrix defined by the representations of sub-image sets instead of pixels. Our method provides a lower-dimensional data representation, which is beneficial for improving the efficiency of classifiers. The experimental results show that the representation obtained by our proposed framework is more discriminative than other methods when performing the task of image set classification. In future, we will consider how to extend our proposed framework to Reproducing Kernel Hilbert Space (RKHS).

6. Acknowledgments

THE PAPER IS SUPPORTED BY THE NATIONAL NATURAL SCIENCE FOUNDATION OF CHINA (GRANT NO 61672265,U1836218), THE 111 PROJECT OF MINISTRY OF EDUCATION OF CHINA (GRANT NO. B12018), UK EPSRC GRANT EP/N007743/1, AND MURI/EPSRC/DSTL GRANT EP/R018456/1.

References

[1] R. Arandjelovic and A. Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[2] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM journal on matrix analysis and applications*, 29(1):328–347, 2007.

[3] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

[4] K.-X. Chen and X.-J. Wu. Component spd matrices: A low-dimensional discriminative data descriptor for image set classification. *Computational Visual Media*, 4(3):245–252, 2018.

[5] K.-X. Chen, X.-J. Wu, R. Wang, and J. Kittler. Riemannian kernel based nyström method for approximate infinite-dimensional covariance descriptors with application to image set classification. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 651–656. IEEE, 2018.

[6] Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350, 2009.

[7] C. Cortes, M. Mohri, and A. Rostamizadeh. Two-stage learning kernel algorithms. In *ICML*, pages 239–246, 2010.

[8] F. Feng, X.-J. Wu, and T. Xu. Object tracking with kernel correlation filters based on mean shift. In *Smart Cities Conference (ISC2), 2017 International*, pages 1–7. IEEE, 2017.

[9] J. Hamm and D. D. Lee. Extended grassmann kernels for subspace-based learning. In *Advances in neural information processing systems*, pages 601–608, 2009.

[10] M. Harandi, M. Salzmann, and R. Hartley. Dimensionality reduction on spd manifolds: The emergence of geometry-aware methods. *IEEE transactions on pattern analysis and machine intelligence*, 40(1):48–62, 2018.

[11] M. Harandi, M. Salzmann, and F. Porikli. Bregman divergences for infinite dimensional covariance matrices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1003–1010, 2014.

[12] Z. Huang and L. Van Gool. A riemannian network for spd matrix learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[13] 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 *International conference on machine learning*, pages 720–729, 2015.

[14] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2012.

[15] T.-K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1415–1428, 2009.

[16] B. Kulis, M. A. Sustik, and I. S. Dhillon. Low-rank kernel learning with bregman matrix divergences. *Journal of Machine Learning Research*, 10(Feb):341–376, 2009.

[17] G. Kylberg, M. Uppström, and I.-M. Sintorn. Virus texture analysis using local binary patterns and radial density profiles. In *Iberoamerican Congress on Pattern Recognition*, pages 573–580. Springer, 2011.

[18] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–409. IEEE, 2003.

[19] P. Li, Q. Wang, W. Zuo, and L. Zhang. Log-euclidean kernels for sparse representation and dictionary learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1601–1608, 2013.

[20] Y. Li, R. Wang, Z. Cui, S. Shan, and X. Chen. Spatial pyramid covariance-based compact video code for robust face retrieval in tv-series. *IEEE Transactions on Image Processing*, 25(12):5905–5919, 2016.

[21] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *European Conference on Computer Vision*, pages 581–595. Springer, 2014.

[22] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *International Journal of computer vision*, 66(1):41–66, 2006.

[23] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European conference on computer vision*, pages 143–156. Springer, 2010.

[24] N. Shroff, P. Turaga, and R. Chellappa. Moving vistas: Exploiting motion for describing scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1911–1918. IEEE, 2010.

[25] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.

[26] S. Sra. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *Advances in neural information processing systems*, pages 144–152, 2012.

[27] H. Sun, X. Zhen, Y. Zheng, G. Yang, Y. Yin, and S. Li. Learning deep match kernels for image-set classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3307–3316, 2017.

[28] H. Tan, Z. Ma, S. Zhang, Z. Zhan, B. Zhang, and C. Zhang. Grassmann manifold for nearest points image set classification. *Pattern Recognition Letters*, 68:190–196, 2015.

[29] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis &amp; Machine Intelligence*, (10):1713–1727, 2008.

[30] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li. Beyond covariance: Feature representation with nonlinear kernel matrices. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4578, 2015.

[31] Q. Wang, P. Li, W. Zuo, and L. Zhang. Raid-g: Robust estimation of approximate infinite dimensional gaussian with application to material recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4433–4441, 2016.

[32] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2496–2503. IEEE, 2012.

[33] R. Wang, X.-J. Wu, K.-X. Chen, and J. Kittler. Multiple manifolds metric learning with application to image set classification. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 627–632. IEEE, 2018.