Approximate Log-Hilbert-Schmidt Distances between Covariance Operators for Image Classification

Hà Quang Minh[1] Marco San Biagio[1] Loris Bazzani[2] Vittorio Murino[1]

[1] Pattern Analysis and Computer Vision (PAVIS), Istituto Italiano di Tecnologia (IIT), Italy
[2] Department of Computer Science, Dartmouth College, USA
{minh.haquang,marco.sanbiagio,vittorio.murino}@iit.it,loris.bazzani@gmail.com

Abstract

This paper presents a novel framework for visual object recognition using infinite-dimensional covariance operators of input features, in the paradigm of kernel methods on infinite-dimensional Riemannian manifolds. Our formulation provides a rich representation of image features by exploiting their non-linear correlations, using the power of kernel methods and Riemannian geometry. Theoretically, we provide an approximate formulation for the Log-Hilbert-Schmidt distance between covariance operators that is efficient to compute and scalable to large datasets. Empirically, we apply our framework to the task of image classification on eight different, challenging datasets. In almost all cases, the results obtained outperform other state of the art methods, demonstrating the competitiveness and potential of our framework.

1. Introduction

Covariance descriptors are a powerful image representation approach in computer vision. In this approach, an image is compactly represented by a covariance matrix encoding correlations between different features extracted from that image. This representation has been demonstrated to work very well in numerous vision tasks, including tracking [24], object detection and classification [31, 30], and image retrieval [7]. Covariance descriptors, properly regularized if necessary, are symmetric positive definite (SPD) matrices, which do not form a vector subspace of Euclidean space under the standard matrix addition and scalar multiplication operations, but form a Riemannian manifold. The optimal measure of similarity between covariance descriptors is thus not the Euclidean distance, but a metric that captures this manifold structure. One of the most commonly used Riemannian metrics in the literature is the Log-Euclidean metric developed by [1]. This is a so-called *bi-invariant Riemannian metric* under which the manifold is flat, that is having zero curvature. Therefore, it is efficient to compute and can be used to define many positive definite kernels, allowing kernel methods to be applied directly on the manifold. This latter property has been exploited successfully in various recent work in vision [15, 19]. However, a major limitation of covariance matrices is that they only capture *linear* correlations between input features.

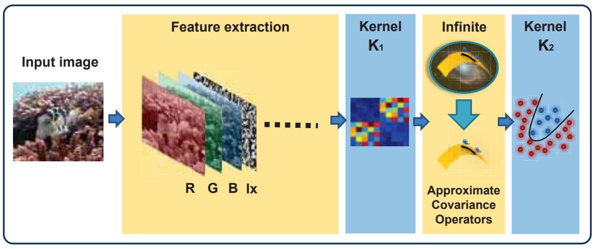

Figure 1. Model of the proposed framework

In this work, we propose to use *infinite-dimensional covariance operators* as image representations. These are covariance matrices of infinite-dimensional features, which are induced implicitly when a positive definite kernel (K_1 in Fig. 1), such as the Gaussian kernel, is applied to the original image features. These covariance operators capture in particular *non-linear correlations* between the original input features. Each image is then represented by one such covariance operator.

For tasks such as image classification, we require the notion of distance between image representations, which in this case means the distance between the corresponding covariance operators. It is known that covariance operators, properly regularized, lie on the *infinite-dimensional Riemannian manifold* of positive definite operators [17]. On this manifold, the generalization of the Log-Euclidean metric is the *Log-Hilbert-Schmidt* (Log-HS) metric [21]. Having computed the Log-HS distances between the covariance operators, another positive definite kernel (K_2 in Fig. 1) can be computed using these distances and used as input to a kernel classifier, e.g. SVM.

In essence, the above two steps, namely image representation by covariance operators and kernel classification, together make up a two-layer kernel machine. The kernel K_1 in the first layer, with the low-level image features as input, induces covariance operators which capture non-linear correlations between these features. The kernel K_2 in the second layer, with the Log-HS distances between covariance operators as input, allows the application of kernel methods, e.g. SVM, to these operators. The result of this double kernelization process is a more powerful representation that can better capture the expressiveness of the image features by exploiting both the power of kernel methods and the Riemannian manifold setting of covariance operators.

However, as with many kernel methods, one drawback of the original Log-HS metric formulation in [21] is that it tends not scale well to large datasets.

To carry out the above kernelization efficiently, we develop the following novel mathematical and computational framework. In this paper, we propose an *approximate Log-HS* distance formulation. This is done by approximating the implicit infinite-dimensional covariance operators above by explicit finite-dimensional covariance matrices, which are computed using explicit approximate feature maps of the original kernel (K_1 in Fig. 1). We demonstrate that the approximate Log-HS distance is substantially faster to compute than the true Log-HS distance, with relatively little loss in the performance of the resulting algorithm. *The main theoretical contribution of the present work is the mathematical derivation and justification of the approximate Log-HS distance, which goes beyond that for kernel approximation.*

In summary, the **novel contributions of our work** are the following. *Mathematically and computationally*, we present an approximate formulation for the Log-HS distance between covariance operators that is substantially faster to compute than the original formulation in [21] and that is scalable to large datasets, while substantially maintaining an effective discriminating capability. *Empirically*, we apply our framework to the task of visual object recognition, using eight challenging, publicly available datatsets, ranging from Fish, Virus, and Texture to Scene recognition. On these datasets, our proposed method compares very favorably with previous state of the art methods in terms of classification accuracy and especially in computational efficiency, demonstrating the competitiveness and potential of our framework.

Related work. Infinite-dimensional covariance operators of low-level features have been applied to the task of image classification recently by [21], which formulated the Log-HS metric, and by [14], which used the formulation for Bregman divergences proposed in [35]. While they both work very well, these methods tend to be computationally intensive and not scalable to large datasets. There exists a large literature on large-scale kernel approximation, which focuses on approximating either the feature maps or the kernel matrices [25, 34, 27], but not on the covariance operators. Approximate affine-invariant distances between covariance operators for image classification have recently been considered in [12]. However, the affine-invariant distance *cannot* be used to define positive definite kernels [15] and, as we show in Sec. 3, this approximation approach is *not* scalable to large datasets.

Organization. We give an overview of the Riemannian distances between finite-dimensional covariance matrices and their generalizations to infinite-dimensional covariance operators in Sec. 2. The core of the paper is Sec. 3, in which we present our approximate Log-HS distance formulation, using two methods for computing approximate feature maps and covariance operators. Empirical results on the task of visual object recognition, using eight different datasets, are reported in Sec. 4. Proofs for all mathematical results are given in the Supplementary Material.

2. Distances between covariance matrices and covariance operators

2.1. Riemannian manifold of SPD matrices

Let $n \in \mathbb{N}$ be fixed. Covariance matrices of size $n \times n$, properly regularized if necessary, are instances of the set $\mathrm{Sym}^{++}(n)$ of SPD matrices, which forms a finite-dimensional Riemannian manifold, see *e.g.* [2, 23]. The most commonly encountered Riemannian metric on $\mathrm{Sym}^{++}(n)$ is the *affine-invariant metric*, in which the geodesic distance between two SPD matrices A and B is

$$d_{\mathrm{aiE}}(A,B) = ||\log(A^{-1/2}BA^{-1/2})||_F, \tag{1}$$

where F denotes the Frobenius norm, which for $A = (a_{ij})_{i,j=1}^n$ is given by $||A||_F^2 = \sum_{i,j=1}^n a_{ij}^2$, and log denotes the principal matrix logarithm. From a practical perspective, the distance (1) tends to be computationally intensive for large scale datasets. This motivated the development of the *Log-Euclidean metric* framework of [1], in which the geodesic distance between A and B is given by

$$d_{\mathrm{logE}}(A,B) = ||\log(A) - \log(B)||_F, \tag{2}$$

This distance is faster to compute than the distance (1), particularly when computing all pairwise distances on a large set of SPD matrices. Furthermore, the Log-Euclidean metric can be used to define many positive definite kernels, such as the Gaussian kernel, which is not possible using the affine-invariant metric [15]. *We wish to emphasize that the Log-Euclidean metric is itself a Riemannian metric on* $\mathrm{Sym}^{++}(n)$*, a so-called bi-invariant metric*, which is a fact not always discussed in recent theoretical work in computer vision, e.g. [13]. This metric arises from the commutative Lie group structure of $\mathrm{Sym}^{++}(n)$, namely $A \odot B = \exp(\log(A) + \log(B))$, introduced by [1], where the corresponding geodesic curves and the geodesic distance (2) are derived. In fact, for two *commuting* matrices A and B, the Log-Euclidean and affine-invariant distances are identical, i.e. $d_{\log\mathrm{E}}(A,B) = d_{\mathrm{aiE}}(A,B)$. One can see that in Eq. (2), the SPD structure of A and B is encoded via the principal matrix logarithm, which is given by $\log(A) = U\mathrm{diag}(\log\lambda_1,\ldots,\log\lambda_n)U^T$, where $A = U\mathrm{diag}(\lambda_1,\ldots,\lambda_n)U^T$ is the spectral decomposition for A (note that if A has a negative eigenvalue, the principal logarithm is *not* defined). In contrast, the standard Euclidean distance $||A-B||_F$ is defined solely in terms of the entries of $A-B$, without reflecting any structure in A and B. Consequently, even though $\mathrm{Sym}^{++}(n)$ has zero curvature under the Log-Euclidean metric, the geodesic distance (2) nevertheless captures better the geometry of $\mathrm{Sym}^{++}(n)$ than the Euclidean distance $||A-B||_F$. This has also been consistently demonstrated empirically, see e.g [1, 15].

The SPD matrices considered in the current work are covariance matrices of features extracted from input data. Specifically, let $\mathcal{X} \subset \mathbb{R}^n$. Let $\mathbf{x} = [x_1,\ldots,x_m]$ be a data matrix sampled from $\mathcal{X}$, where m is the number of observations. In the setting of the current work, there is one such matrix for each image, namely the matrix of low-level features sampled at (a subset of) the pixels in the image. Each image is then represented by the $n\times n$ covariance matrix

$$C_{\mathbf{x}} = \frac{1}{m}\mathbf{x}J_m\mathbf{x}^T : \mathbb{R}^n \to \mathbb{R}^n, \tag{3}$$

where J_m is the centering matrix, defined by $J_m = I_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T$ with $\mathbf{1}_m = (1,\ldots,1)^T \in \mathbb{R}^m$.

In practice, $C_{\mathbf{x}}$ is generally only positive semi-definite and thus to apply the Riemannian structure of $\mathrm{Sym}^{++}(n)$, it is often necessary to consider the regularized version $(C_{\mathbf{x}} + \gamma I_n)$ for some $\gamma > 0$. For two covariance matrices $C_{\mathbf{x}}$ and $C_{\mathbf{y}}$, we therefore consider the distance between the regularized versions $(C_{\mathbf{x}} + \gamma I)$ and $(C_{\mathbf{y}} + \mu I)$, given by

$$d_{\log\mathrm{E}} = ||\log(C_{\mathbf{x}} + \gamma I_n) - \log(C_{\mathbf{y}} + \mu I_n)||_F, \tag{4}$$

for some regularization parameters $\gamma > 0$, $\mu > 0$.

2.2. Infinite-dimensional covariance operators

The covariance matrix $C_{\mathbf{x}}$ only measures the *linear* correlations between the features in the input data. A powerful method to capture *non-linear* input correlations is by (i) first mapping the original input features into a high dimensional feature space $\mathcal{H}$, using an implicit nonlinear feature map induced by a positive definite kernel; (ii) then computing the covariance operators in the feature space $\mathcal{H}$.

Specifically, let $\mathcal{X}$ be an arbitrary non-empty set. Let $\mathbf{x} = [x_1,\ldots,x_m]$ be a data matrix sampled from $\mathcal{X}$, where $m \in \mathbb{N}$ is the number of observations. Let K be a positive definite kernel on $\mathcal{X}\times\mathcal{X}$ and $\mathcal{H}_K$ its induced reproducing kernel Hilbert space (RKHS). Let $\mathcal{H}$ be any feature space for K, which we assume to be a separable Hilbert space, with the corresponding feature map $\Phi : \mathcal{X} \to \mathcal{H}$, so that $K(x,y) = \langle\Phi(x),\Phi(y)\rangle_{\mathcal{H}}$ for all pairs $(x,y) \in \mathcal{X}\times\mathcal{X}$. For concreteness, we can identify $\mathcal{H}$ with the RKHS $\mathcal{H}_K$, or with the space of square summable sequences $\ell^2 = \{(a_k)_{k\in\mathbb{N}} : \sum_{k=1}^{\infty}|a_k|^2 < \infty\}$. The feature map Φ gives the (potentially infinite) mapped data matrix $\Phi(\mathbf{x}) = [\Phi(x_1),\ldots,\Phi(x_m)]$ of size $\dim(\mathcal{H})\times m$ in the feature space $\mathcal{H}$. The corresponding covariance operator for $\Phi(\mathbf{x})$ is defined to be

$$C_{\Phi(\mathbf{x})} = \frac{1}{m}\Phi(\mathbf{x})J_m\Phi(\mathbf{x})^T : \mathcal{H} \to \mathcal{H}, \tag{5}$$

which can be considered as a (potentially infinite) matrix of size $\dim(\mathcal{H})\times\dim(\mathcal{H})$. If $\mathcal{X} = \mathbb{R}^n$ and $K(x,y) = \langle x,y\rangle$, then $C_{\Phi(\mathbf{x})} = C_{\mathbf{x}}$ as in (3).

Infinite-dimensional affine-invariant metric. As in the finite-dimensional case, we need to consider the regularized covariance operator $(C_{\Phi(\mathbf{x})} + \gamma I_{\mathcal{H}})$, $\gamma > 0$, which lies on the infinite-dimensional manifold $\Sigma(\mathcal{H})$ of positive definite operators on $\mathcal{H}$. Let $A : \mathcal{H} \to \mathcal{H}$ be a bounded linear operator and A^* be its adjoint operator. We recall that $A : \mathcal{H} \to \mathcal{H}$ is said to be a Hilbert-Schmidt operator, denoted by $A \in \mathrm{HS}(\mathcal{H})$, if

$$||A||^2_{\mathrm{HS}} = \mathrm{tr}(A^*A) = \sum_{k=1}^{\infty}\lambda_k(A^*A) < \infty, \tag{6}$$

where $||\ ||_{\mathrm{HS}}$ denotes the Hilbert-Schmidt norm, the infinite-dimensional generalization of the Frobenius norm, and $\{\lambda_k(A^*A)\}_{k=1}^{\infty}$ denotes the set of eigenvalues of A^*A. By the formulation of [17, 22], the infinite-dimensional affine-invariant distance d_{aiHS} between $(C_{\Phi(\mathbf{x})} + \gamma I)$ and $(C_{\Phi(\mathbf{y})} + \mu I)$ is given by

$$\begin{aligned} &d_{\mathrm{aiHS}}[(C_{\Phi(\mathbf{x})} + \gamma I), (C_{\Phi(\mathbf{y})} + \mu I)] \qquad (7)\\ &= ||\log[(C_{\Phi(\mathbf{x})} + \gamma I)^{-1/2}(C_{\Phi(\mathbf{y})} + \mu I)(C_{\Phi(\mathbf{x})} + \gamma I)^{-1/2}]||_{\mathrm{eHS}}, \end{aligned}$$

with the extended Hilbert-Schmidt norm $||\ ||_{\mathrm{eHS}}$ given by $||A + \gamma I||^2_{\mathrm{eHS}} = ||A||^2_{\mathrm{HS}} + \gamma^2$.

Log-Hilbert-Schmidt metric. The generalization of the Log-Euclidean metric to the infinite-dimensional manifold $\Sigma(\mathcal{H})$ has recently been formulated by [21]. In this metric, termed *Log-Hilbert-Schmidt metric*, or *Log-HS* for short, the distance $d_{\log\mathrm{HS}}[(C_{\Phi(\mathbf{x})} + \gamma I_{\mathcal{H}}), (C_{\Phi(\mathbf{y})} + \mu I_{\mathcal{H}})]$ between $(C_{\Phi(\mathbf{x})} + \gamma I_{\mathcal{H}})$ and $(C_{\Phi(\mathbf{y})} + \mu I_{\mathcal{H}})$ is given by

$$\begin{aligned} &d_{\log\mathrm{HS}}[(C_{\Phi(\mathbf{x})} + \gamma I_{\mathcal{H}}), (C_{\Phi(\mathbf{y})} + \mu I_{\mathcal{H}})] \qquad (8)\\ &= ||\log(C_{\Phi(\mathbf{x})} + \gamma I_{\mathcal{H}}) - \log(C_{\Phi(\mathbf{y})} + \mu I_{\mathcal{H}})||_{\mathrm{eHS}}, \end{aligned}$$

which has a closed form in terms of the corresponding Gram matrices (we refer to [21] for the explicit expression).

Regularization. The form of regularization $(A + \gamma I)$, $\gamma > 0$, which is often *empirically* necessary to ensure positive definiteness in the case $\dim(\mathcal{H}) < \infty$, is always necessary, both *theoretically and empirically*, when $\dim(\mathcal{H}) = \infty$, since in this case $\log(A)$, with A being a covariance operator, is always unbounded (see [17, 21, 22]).

As in the finite-dimensional case, *two key advantages of the Log-HS distance* $d_{\log\mathrm{HS}}$ over the affine-invariant distance d_{aiHS} are: (i) the $d_{\log\mathrm{HS}}$ distance is faster to compute than the d_{aiHS} distance; (ii) it is straightforward to define many commonly used positive definite kernels, such as the Gaussian kernel, using the $d_{\log\mathrm{HS}}$ distance, which is not the case with the d_{aiHS} distance. These advantages are fully exploited in the current work.

In the next section, we describe how to approximate Formula (8) to compute the pairwise distances on a set of data matrices $\{\mathbf{x}_i\}_{i=1}^N$ when N and m are large.

3. Approximate Log-HS Distance

While kernel methods are powerful in learning nonlinear structures in data, they tend not to scale well, in terms of computational complexity, to large datasets, which are common in vision problems such as object recognition and fine-grained categorization. In the typical kernel learning setting, the feature map Φ is high-dimensional (and often infinite-dimensional, as in the case of the Gaussian kernel) and thus is only used *implicitly*. Instead, exact kernel methods carry out computations using Gram matrices and thus their computational complexities depend on the sizes of the Gram matrices, which become very large for large datasets.

A commonly used approach that has emerged recently to reduce the computational cost of kernel methods is to compute an *explicit approximate feature map* $\hat{\Phi}_D : \mathcal{X} \to \mathbb{R}^D$, where D is finite and $D << \dim(\mathcal{H})$, so that

$$\langle \hat{\Phi}_D(x), \hat{\Phi}_D(y) \rangle_{\mathbb{R}^D} = \hat{K}_D(x,y) \approx K(x,y), \quad \text{with} \tag{9}$$

$$\lim_{D\to\infty} \hat{K}_D(x,y) = K(x,y) \quad \forall (x,y) \in \mathcal{X} \times \mathcal{X}. \tag{10}$$

The approximate feature map $\hat{\Phi}_D$ is then used directly in the learning algorithms instead of the Gram matrices.

In our setting, we use the approximate feature maps to compute the corresponding finite-dimensional approximate covariance operators. The Log-Euclidean distance between the approximate operators is then used as the approximate version of Log-HS distance between the infinite-dimensional covariance operators. *Thus we are not interested in the approximate kernel values $\hat{K}_D(x,y)$ per se, but the approximate covariance operators and the corresponding approximate Log-HS distance. The mathematical justification for the latter goes beyond that for kernel approximation and is the main theoretical contribution of this work.* In fact, as we show in Theorems 1 and 2 below, Eq. (10), which guarantees the convergence of the approximate kernel value to the true kernel value, is *not sufficient* to guarantee the convergence of the approximate Log-HS distance to the true Log-HS distance as $D \to \infty$. This convergence is non-trivial and requires further assumptions, which are practically realizable.

Approximate covariance operator and approximate Log-HS distance. With the approximate feature map $\hat{\Phi}_D$, we have the corresponding data matrix $\hat{\Phi}_D(\mathbf{x}) = [\hat{\Phi}_D(x_1), \ldots, \hat{\Phi}_D(x_m)]$ of size $D \times m$, and the approximate covariance operator has the form

$$C_{\hat{\Phi}_D(\mathbf{x})} = \frac{1}{m}\hat{\Phi}_D(\mathbf{x}) J_m \hat{\Phi}_D(\mathbf{x})^T : \mathbb{R}^D \to \mathbb{R}^D, \tag{11}$$

which is a matrix of size $D \times D$, instead of the potentially infinite matrix $C_{\Phi(\mathbf{x})}$ of size $\dim(\mathcal{H}) \times \dim(\mathcal{H})$.

We then consider the following as an approximate version of the Log-HS distance given in Formula (8):

$$\left\| \log\left(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I_D\right) - \log\left(C_{\hat{\Phi}_D(\mathbf{y})} + \mu I_D\right)\right\|_F. \tag{12}$$

Key theoretical question. We need to determine whether Formula (12) is truly a finite-dimensional approximation of Formula (8), in the sense that

$$\lim_{D\to\infty} \left\| \log(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I_D) - \log(C_{\hat{\Phi}_D(\mathbf{y})} + \mu I_D)\right\|_F$$
$$= ||\log(C_{\Phi(\mathbf{x})} + \gamma I_{\mathcal{H}}) - \log(C_{\Phi(\mathbf{y})} + \mu I_{\mathcal{H}})||_{\mathrm{eHS}}. \tag{13}$$

The answer to this question is the main mathematical contribution of the current paper. It turns out that in general, this is *not* possible. This is because the infinite-dimensional Log-HS distance is generally *not* obtainable as a limit of the finite-dimensional Log-Euclidean distance as the dimension approaches infinity [21]. More precisely, we have

Theorem 1 *Assume that $\gamma \neq \mu$, $\gamma > 0$, $\mu > 0$. Then*

$$\lim_{D\to\infty} \left\| \log(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I_D) - \log(C_{\hat{\Phi}_D(\mathbf{y})} + \mu I_D)\right\|_F = \infty.$$

The infinite limit in Theorem 1 stands in sharp contrast to that of Eq. (10) on the approximability of the kernel value $K(x,y)$ itself, which is satisfied by both approximation schemes based on Fourier features presented below.

In practice, however, it is reasonable to assume that we can use the same regularization parameter for both $C_{\hat{\Phi}_D(\mathbf{x})}$ and $C_{\hat{\Phi}_D(\mathbf{y})}$, that is to set $\gamma = \mu$. In this setting, we obtain the necessary convergence, as follows.

Theorem 2 *Assume that $\gamma = \mu > 0$. Then*

$$\lim_{D\to\infty} \left\| \log(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I_D) - \log(C_{\hat{\Phi}_D(\mathbf{y})} + \gamma I_D)\right\|_F$$
$$= ||\log(C_{\Phi(\mathbf{x})} + \gamma I_{\mathcal{H}}) - \log(C_{\Phi(\mathbf{y})} + \gamma I_{\mathcal{H}})||_{\mathrm{eHS}}. \tag{14}$$

In light of Theorems 1 and 2, subsequently we employ the same regularization parameter $\gamma > 0$ to compute approximate Log-HS distances between all regularized operators $(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I_D)$. In this work, we employ two methods for computing the approximate feature map $\hat{\Phi}_D$, namely Random Fourier features [25] and Quasi-random Fourier features [34], presented in the following section.

3.1. Fourier feature maps

Random Fourier feature maps. This is the approach in [25] for computing approximate feature maps of shift-invariant kernels. Let $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a kernel of the form $K(x,y) = k(x-y)$ for some positive definite function k on $\mathbb{R}^n$. By Bochner's Theorem [26], there is a finite positive measure ρ on $\mathbb{R}^n$ such that

$$k(x-y) = \int_{\mathbb{R}^n} e^{-i\langle \omega, x-y\rangle} d\rho(\omega) \tag{15}$$
$$= \int_{\mathbb{R}^n} \rho(\omega)\phi_\omega(x)\overline{\phi_\omega(y)} d\omega, \text{ where } \phi_\omega(x) = e^{-i\langle \omega, x\rangle}.$$

Without loss of generality, we can assume that ρ is a probability measure on $\mathbb{R}^n$, so that $K(x,y) = \mathbb{E}_{\omega\sim\rho}[\phi_\omega(x)\overline{\phi_\omega(y)}]$. By symmetry, $K(x,y) = \frac{1}{2}[K(x,y) + K(y,x)]$, so that by the relation $\frac{1}{2}[e^{-i\langle\omega,x-y\rangle} + e^{i\langle\omega,x-y\rangle}] = \cos(\langle\omega, x-y\rangle)$ we have

$$K(x, y) = \int_{\mathbb{R}^n} \cos(\langle \omega, x - y \rangle)\rho(\omega) d\omega. \quad (16)$$

To approximate $K(x, y)$, we can sample D points $\{\omega_j\}_{j=1}^D$ from the distribution ρ and compute the empirical version

$$\hat{K}_D(x, y) = \frac{1}{D}\sum_{j=1}^{D} \cos(\langle \omega_j, x - y \rangle) \xrightarrow{D \to \infty} K(x, y) \quad (17)$$

almost surely by the law of large numbers. Let $W = (\omega_1, \ldots, \omega_D)$ be a matrix of size $n \times D$, with each column $\omega_j \in \mathbb{R}^n$ randomly sampled according to ρ. Motivated by the cosine addition formula, $\cos(\langle \omega_j, x - y \rangle) = \cos\langle \omega_j, x \rangle \cos\langle \omega_j, y \rangle + \sin\langle \omega_j, x \rangle \sin\langle \omega_j, y \rangle$, we define

$$\cos(W^T x) = (\cos\langle \omega_j, x \rangle)_{j=1}^D, \quad (18)$$

$$\sin(W^T x) = (\sin\langle \omega_j, x \rangle)_{j=1}^D. \quad (19)$$

The desired approximate feature map is the concatenation

$$\hat{\Phi}_D(x) = \frac{1}{\sqrt{D}}(\cos(W^T x); \sin(W^T x)) \in \mathbb{R}^{2D}, \quad (20)$$

with $\langle \hat{\Phi}_D(x), \hat{\Phi}_D(y) \rangle = \hat{K}_D(x, y)$. In the case of the Gaussian kernel $K(x, y) = e^{-\frac{||x-y||^2}{\sigma^2}}$ (used in the experiments in Sec. 4), we have

$$\rho(\omega) = \frac{(\sigma\sqrt{\pi})^n}{(2\pi)^n} e^{-\frac{\sigma^2 ||\omega||^2}{4}} \sim \mathcal{N}\left(0, \frac{2}{\sigma^2}\right). \quad (21)$$

Quasi-random Fourier feature maps. The Random Fourier feature maps above arise from the Monte-Carlo approximation of the kernel K expressed as the integral in Eq. (15), using a *random* set of points ω_j's sampled according to the distribution ρ. An alternative approach, proposed by [34], employs Quasi-Monte Carlo integration [9], in which the ω_j's are *deterministic* points arising from a *low-discrepancy* sequence in $[0, 1]^n$. We describe this approach in detail in the Supplementary Material.

3.2. New positive definite kernels using approximate Log-HS distances

In our framework, starting with a shift-invariant kernel K_1 in Fig. 1, we compute the approximate feature map $\hat{\Phi}_D(x)$ using the methods in Sec. 3.1 and the corresponding approximate covariance operators using Eq. (11). The approximate Log-HS distances between these approximate covariance operators are then computed using Eq. (12).

With the approximate Log-HS distances, we can define a new positive definite kernel (K_2 in Fig. 1), for example

$$\exp(-|| \log(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I_D) - \log(C_{\hat{\Phi}_D(\mathbf{y})} + \gamma I_D)||_F^p / \sigma^2), \quad (22)$$

for $0 < p \leq 2$, with $p = 2$ giving the Gaussian kernel and $p = 1$ giving the Laplacian kernel. This new kernel can then be used in a classifier, e.g. SVM. The complete pipeline for our framework is summarized in Algorithm 1.

3.3. Computational complexity

We present here the computational analysis of the proposed approximation in Eq. (12) as well as the comparison with the approximate affine-invariant distance proposed by [12], according to the formula

$$\left|\left|\log[(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I)^{-1/2}(C_{\hat{\Phi}_D(\mathbf{y})} + \mu I)(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I)^{-1/2}]\right|\right|_F. \quad (23)$$

The main computational cost in Eq. (12) is the SVD for $(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I)$ and $(C_{\hat{\Phi}_D(\mathbf{y})} + \mu I)$, which takes time $O(D^3)$. At first glance, the computational complexity for the approximate affine-invariant distance in Eq. (23), which consists of a matrix square root and inversion, two matrix multiplications and an SVD, is also $O(D^3)$. However, computationally, the key difference between Eq. (12) and Eq. (23) is that in Eq. (12), $(C_{\hat{\Phi}_D(\mathbf{x})} + \gamma I)$ and $(C_{\hat{\Phi}_D(\mathbf{y})} + \mu I)$ are *uncoupled*, whereas in Eq. (23), they are *coupled*. Thus if we have N data matrices $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, to compute their pairwise approximate Log-HS distances using Eq. (12), we need to compute an SVD for each $(C_{\hat{\Phi}_D(\mathbf{x}_i)} + \gamma I)$, with time complexity $O(ND^3)$. On the other hand, to compute their pairwise approximate affine-invariant distances using Eq. (23), we need to compute an SVD for each *pair* $(C_{\hat{\Phi}_D(\mathbf{x}_i)} + \gamma I)$, $(C_{\hat{\Phi}_D(\mathbf{x}_j)} + \gamma I)$, with time complexity $O(N^2 D^3)$. Thus the approximation of the Log-HS distance is $O(N)$ times faster than the approximation of the affine-invariance distance.

We also note that for N pairs of data matrices, the computational complexity of the exact Log-HS formulation [21] and the RKHS Bregman divergences [14] is of order $O(N^2 m^3)$. Thus for $D < m$ and N large, the approximate Log-HS formulation will be much more efficient to compute than both the exact Log-HS and the RKHS Bregman divergences (see the actual running time comparison between the approximate and exact Log-HS formulations in the experiments below).

Input: Set of images.
Output: Kernel matrix (used as input to a classifier, e.g. SVM).
Parameters: Kernels K_1, K_2, regularization parameters $\gamma = \mu > 0$, approximate feature dimension D.
Procedure:

1. For each image, extract a data matrix $\mathbf{x} = [x_1, \ldots, x_m]$ of low-level features from m pixels.
2. For each image, compute the approximate feature maps $\hat{\Phi}_D(x_i)$, $1 \leq i \leq m$, associated to kernel K_1, according to Eq. (20), and the corresponding approximate covariance operator $C_{\hat{\Phi}_D(\mathbf{x})}$, according to Eq. (11).
3. For each pair of images, compute the approximate Log-HS distance between the corresponding covariance operators, according to Eq. (12).
4. Using kernel K_2, compute a kernel matrix using the above approximate Log-HS distance, e.g. according to Eq. (22).

Algorithm 1: Summary of the proposed method.

Further comparison with [12]. In [12], the authors proposed two methods: (i) Nearest Neighbor using effectively the approximate affine-invariant distance given in Eq.(23) with $\gamma = \mu = 0$ and (ii) the CDL algorithm using the representation $\log(C_{\hat{\Phi}_D(\mathbf{x})})$. Both of these methods require the assumption that $C_{\hat{\Phi}_D(\mathbf{x})}$ is positive definite, which is *never* guaranteed. In fact, when $D > m$, $C_{\hat{\Phi}_D}(\mathbf{x})$ is always rank-deficient and neither its inverse nor $\log$ can be computed. Thus neither the CDL nor the approximate affine-invariant distance can be used. Theoretically, since it does not employ any regularization, the approximate affine-invariant distance in [12] will *not* approach the exact affine-invariant distance ([17, 22]) for large D, which *always* requires regularization (see Sec. 2.2).

4. Experimental results

In this section, we show the performance of the proposed method compared with other state-of-the-art approaches on eight image classification datasets. In all the datasets, we used the same features and experimental protocols of the compared state-of-the-art approaches (see details below). The following methods were evaluated: *LogE*, using the Log-Euclidean metric, *Stein*, using the Stein (also called Jensen-Bregman LogDet) divergence [7], *Log-HS*, using the Log-HS metric [21] induced by the Gaussian kernel (K_1 in Fig. 1, but only on the Fish dataset, see below), and *Approx LogHS* and *QApprox LogHS* induced by the Gaussian kernel, using the proposed random Fourier and Quasi-random Fourier approximation methods in Sec. 3, respectively. For a good trade-off between speed and accuracy, we set the approximate feature dimension $D = 200$ (Eq. 20) for *Approx LogHS* and *QApprox LogHS* (more details below). Except with *Stein*, all experiments used LIBSVM [6] for classification, with the Gaussian kernel defined on top of the corresponding metric (K_2 in Fig. 1). For *Stein*, since the corresponding Gaussian kernel is generally not guaranteed to be positive definite [28], we used the Nearest Neighbor approach as in [7]. We also compared the above methods with CDL [32], one of the state-of-the-art approaches in covariance-based learning. On the Fish dataset, we also carried out experiments with the *Euclidean (E)* and *Hilbert-Schmidt (HS)* metrics to demonstrate the advantage of the Riemannian geometric framework[1]. The performance is evaluated in terms of classification accuracy (more details below). All parameters were chosen by cross-validation.

For each image, at the pixel location (x, y), the following image features were extracted (the actual features vary between the different datasets, since for fairness we followed the same protocols of the methods we compared with)

$$F(x,y) = [x, y, I(x,y), |I_x|, |I_y|, |I_{xx}|, |I_{yy}|, \\ R(x,y), G(x,y), B(x,y), |G_{x,y}^{(o,s)}|] \quad (24)$$

where $I, I_x, I_y, I_{xx}, I_{yy}$ denote the intensity and its first- and second-order derivatives, respectively, R,G, and B denote the color values, and $G_{x,y}^{o,s}$ denotes the Gabor filter at orientation o and scale s.

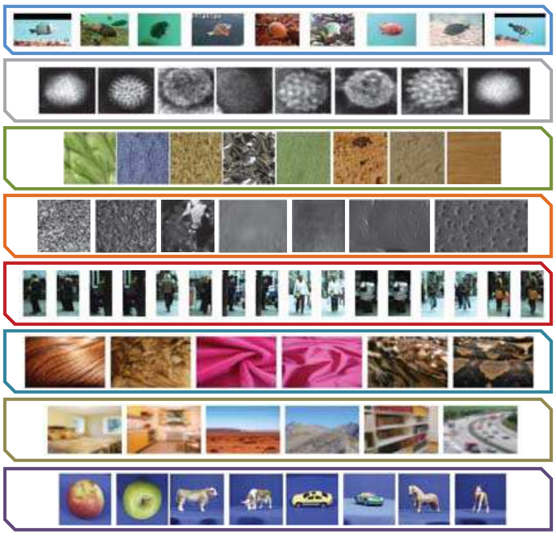

Figure 2. Sample images for datasets used in this work. From top to bottom: Fish Recognition [3], Virus Classification [16], KTH-TIPS2b Material [5], Texture [4, 8], ETHZ [10], UIUC [20], Tiny-Graz03 [33], and ETH80 [18].

4.1. Datasets

The **Fish recognition dataset** [3] contains 27,370 fish images acquired from live video. There are altogether 23 classes of fish and the number of images per class ranges from 21 to 12,112, with a mean resolution of 150 × 120 pixels. The significant variations in color, pose and illumination inside each class make this dataset very challenging. We conducted two different experiments on this dataset, using the R,G,B features from Eq. (24), as in [21]. The first experiment (named Exp1) is a small scale experiment which used the same protocol and the 10 splits provided by [21], consisting altogether of 345 images, divided in 115 images for training (5 per class) and 230 images for testing (10 per class) . We analyzed the performances of *Approx LogHS* and *QApprox LogHS* with respect to the original Log-HS metric formulation of [21]. Furthermore, we compared the computational cost and running time of the two approximate versions with respect to [21]. In the second experiment (named Exp2), the entire Fish dataset was used to show the scalability of the proposed method. We note that this experiment was not carried out using the exact Log-HS metric due to its high computational cost in terms of memory and speed. In this case, since the number of testing samples are different between the classes, the classification performance is evaluated using the Average Precision (AP) measure, a standard metric used by the PASCAL challenge [11].

The **Virus Classification dataset** [16] contains 15 different virus classes. Each class has 100 images of size

[1] Due to lack of space, the experimental results for the *E* and *HS* metrics on the other datasets will be reported in the longer version of the paper.

41×41. For this dataset, following [14], we employed the 25-dimensional feature vector consisting of the intensity, 4 gradients, and 20 Gabor filters in Eq. (24) at 4 orientations and 5 scales. We used the 10 splits provided by the authors in a leave-one-out manner, *i.e.* 9 splits for training and 1 split as query, repeating the procedure 10 times.

The **KTH-TIPS2b Material dataset** [5] contains images of 11 materials captured under 4 illuminations, in 3 poses and at 9 scales, giving 108 images for each sample in a category, with 4 samples per material. For each category, we trained on 3 samples and tested on the remaining sample. For this dataset, following [14], we extracted the 23-dimensional feature consisting of the R,G,B values and 20 Gabor filters in Eq. (24) at 4 orientations and 5 scales.

The **Texture dataset** for our experiments was created by combining 111 texture images of the Brodatz dataset [4] and 61 of the CURET dataset [8], as done in [7]. Unfortunately, we were not able to reproduce the experiments in [7] since the exact protocols, i.e. the number of patches extracted from each image and the number of training/testing images, were not specified. We therefore carried out a similar experiment by extracting 150 patches of size 20×20 from each image, taking 140 as training and 10 as testing, repeating the entire procedure 10 times. For this dataset, we extracted the 5-dimensional feature vector $[x, y, I, |I_x|, |I_y|]$ as in [7].

For person re-identification, we used two sequences of the **ETHZ dataset** [10]. SEQ. #1 contains 83 pedestrians in $4,857$ images. SEQ. #2 contains 35 pedestrians in $1,936$ images. As in [14], we used 10 images from each subject for training and the rest for testing. Following [14], we extracted the 17-dimensional feature vector consisting of $[x, y, R, G, B]$ and the first- and second-order color derivatives, i.e. $[|\frac{\partial r}{\partial x}|, |\frac{\partial r}{\partial y}|, |\frac{\partial^2 r}{\partial x^2}|, |\frac{\partial^2 r}{\partial y^2}|]$, for $r = R, G, B$. The performance was evaluated using the Average Precision metric.

The **UIUC dataset** [20] contains 18 different material categories collected *in the wild*, with a total of 216 images. Following [12], we extracted the 19-dimensional vector consisting of 3 colors, 4 gradients, and 12 Gabor filters in Eq.(24) at 4 orientations and 3 scales. As in [12], we split the database into training and test sets by randomly assigning half of the images of each class to the training set and testing on the rest. This process was repeated 10 times.

The **TinyGraz03 dataset** [33] contains 1148 indoor and outdoor scenes with an image resolution of 32×32 pixels. The images are divided in 20 classes with at least 40 samples per class. We used the recommended train/test split provided by the authors. For this dataset, following [12], at each pixel we extracted the 7-dimensional feature vector $[|I_x|, |I_y|, |I_{xx}|, |I_{yy}|, R, G, B]$ from Eq. (24). This dataset is highly challenging and the correct recognition rate achieved by humans is only 30% [33].

The **ETH80 dataset** [18] contains images of eight object categories: apples, cows, cups, dogs, horses, pears, tomatoes, and cars. Each category includes ten object subcategories (e.g. various dogs) in 41 orientations, resulting in 410 images per category. We randomly chose 21 images for training and the rest for testing, repeating the procedure 10 times. For this dataset, following [15], at each pixel we extracted the 5-dimensional feature vector $[x, y, I(x, y), |I_x|, |I_y|]$ from Eq. (24).

Method	Accuracy Exp1	Accuracy Exp2
Approx LogHS	$\mathbf{53.91}\%(\pm 4.34)$	$\mathbf{56.2}\%(\pm 2.2)$
QApprox LogHS	$\mathbf{54.30}\%(\pm 3.44)$	$\mathbf{57.70}\%(\pm 1.8)$
LogHS [21]	$56.74\%(\pm 2.87)$	N/A
HS	$50.17\%(\pm 2.17\%)$	$52.49\%(\pm 2.26\%)$
LogE	$42.70\%(\pm 3.45)$	$46.20\%(\pm 1.9)$
E	$26.87\%(\pm 3.52\%)$	$28.18\%(\pm 1.72\%)$
Stein [7]	$43.95\%(\pm 4.48)$	$40.83\%(\pm 7.5)$
CDL [32]	$41.70\%(\pm 3.60)$	$42.8\%(\pm 2.0)$

Table 1. Results on the Fish dataset [3] in terms of classification accuracy. Two different experiments were conducted: Exp1 compares *Approx LogHS* and *QApprox LogHS* with respect to the original Log-HS metric on the reduced dataset. For Exp2, the results are reported for the whole dataset.

4.2. Analysis and discussion of results

Classification performance compared with exact Log-HS metric. First of all, we ran a comparative experiment on a subset of the Fish dataset [3], as in [21], to analyze the performances of the *Approx LogHS* and *QApprox LogHS* formulations with respect to the original Log-HS formulation of [21]. For this dataset, we show results obtained using R,G,B features as in [21] with all methods. Column two of Tab.1 shows the mean and standard deviation values for the classification accuracies computed over all 10 random splits. The first important observation we note is that *Approx LogHS* and *QApprox LogHS* gave results (first and second rows) which are comparable with those using the Log-HS metric and substantially better than other methods, namely *LogE*, *Stein*, and *CDL*. Furthermore, the Riemannian distance *LogE* substantially outperforms the Euclidean distance *E*, and similarly *LogHS*, *Approx LogHS*, and *QApprox LogHS* all outperform the Hilbert-Schmidt distance *HS*. This clearly demonstrates the advantage of the Riemannian geometric framework.

Running time compared with exact Log-HS metric. Most importantly, *Approx LogHS* and *QApprox LogHS* incur much smaller computation costs compared to Log-HS. In fact, using MATLAB on an Intel Xeon E5-2650, 2.60 GHz PC, we obtained a speed up of $30\times$ with *QApprox LogHS* (Train: 6.7sec. Test: 18sec.) and more than $50\times$ with *Approx LogHS* (Train: 3.6sec. Test: 9.9sec.) with respect to the baseline Log-HS (Train: 175.7sec. Test: 565.1sec.). Because of this substantial speed up in running time, subsequently we focused solely on the performance of *Approx LogHS* and *QApprox LogHS* for all the other datasets.

Method	**Virus** Acc %	**KTH-TIPS2b** Acc %	**Texture** Acc %	**ETHZRe-ID** Acc-Seq1 %	 Acc-Seq2 %	**UIUC** Acc %	**TinyGraz03** Acc %	**ETH80** Acc %
Approx LogHS	81.5% (±2.1)	**83.6%** (±5.4)	**76.9%** (±0.5)	**92.0%** (±0.3)	**93.2%** (±0.5)	**50.1%** (±3.7)	**60%**	**95.0%** (±0.5)
QApprox LogHS	76.5% (±3.2)	83.46% (±5.6)	76.4% (±0.6)	91.9% (±0.5)	93.0% (±0.5)	44.7% (±3.6)	57%	94.9% (±0.6)
LogE	71.9% (±4.0)	74.1% (±7.4)	52.9% (±0.8)	89.9% (±0.2)	91.9% (±0.4)	37.8% (±2.6)	40%	71.1% (±1.0)
Stein [7]	49.7% (±4.8)	73.1% (±8.0)	38.4% (±0.7)	89.6% (±0.8)	90.9% (±0.2)	27.9% (±1.7)	24%	67.5% (±0.4)
CDL [32]	69.5% (±3.1)	76.3% (±5.11)	53.8% (±0.5)	86.8% (±0.6)	88.8% (±1.2)	36.3% (±2.0)	41%	56.0% (±0.6)
SoA	**82.5%** (±2.9) [12]	80.1% (±4.6) [14]	N/A	90.2% (±1.0) [14]	91.4% (±0.8) [14]	47.4% (±3.1) [12]	57% [12]	83.6% (±6.1) [29]

Table 2. Best results obtained on seven different datasets in terms of classification accuracy. The first two rows represent the proposed method using the two-layer kernel machine using *Approx LogHS* and *QApprox LogHS* with Gaussian SVM. The last row represents the state-of-the-art results on each dataset.

With this computational speed-up, we next ran a second experiment (third column) using the whole Fish dataset. The classification accuracy shows an improvement of 10% and 11.5% for *Approx LogHS* and *QApprox LogHS*, respectively, with respect to *LogE* and an improvement of 15.4% and 16.9% with respect to the *Stein* divergence [7].

Comparison against other state of the art (SoA) methods. Table 2 shows the results of the proposed method (first two rows) on seven different datasets in comparison with the respective state-of-the-art results. The best result [12] reported for the Virus dataset is an accuracy of 82.5% (last row). Our classification accuracy is slightly lower than this result but it outperforms all the other competitors (*LogE, Stein and CDL*, by 9.6%, 31.8%, and 12%, respectively). Our results on the KTH-TIPS2b Material dataset improves the state of the art [14] by 3.5% (third column). The accuracy of the proposed method on the Texture dataset (forth column) is 23.1% higher of the best of the other competitors. The fifth and sixth columns of Table 2 report the results on two sequences of the ETHZRe-ID dataset. In this case, we obtained an improvement of 1.8% over the state-of-the-art [14]. Regarding the UIUC and the TinyGraz03 datasets, the improvement over the previous results of [12], is 2.7% and 3%, respectively. Our method outperforms the recent state of the art [29] also on the ETH80 dataset with an improvement of 11.4%.

These improvements in classification accuracies all demonstrate the effectiveness of our proposed method. *More importantly, we emphasize that our approximate Log-HS formulations are much more computationally efficient than both the RKHS Bregman divergences in [14] and the approximate affine-invariant distance used in [12], as discussed in Sec. 3.3.*

Increasing the approximate feature dimension. We also carried out a set of experiments to show that the classification accuracy improves when increasing the approximate feature dimension D in Eq. (20), as expected. We tested this on the TinyGraz03 dataset, with $D = 200, 400, 1000$, with the results reported in Tab. 3. However, as also expected, the improvement in classification accuracy comes at a larger computational cost. In fact, the training time for $D = 1000$ is 63 and 48 times slower than $D = 200$ for *Approx LogHS* and *QApprox LogHS*, respectively. In other words, there is a very large computational speed up factor from $D = 1000$ to $D = 200$ with a very low drop in accuracy.

Method	**D = 200**	**D = 400**	**D = 1000**
Approx LogHS	57% $3.8s$	59% $20.5s$	60% $240.4s$
QApproxLogHS	55% $5.6s$	58% $26.7s$	59% $271.7s$

Table 3. Results on the TinyGraz03 dataset [33] with increasing values of the approximate feature dimension D. We also reported the training time in seconds below each accuracy.

5. Conclusion, discussion, and future work

In this paper, we have presented a novel mathematical and computational framework for visual object recognition using infinite-dimensional covariance operators of image features, in the paradigm of kernel methods and Riemannian geometry. Our approximate Log-HS distance formulation is substantially faster to compute than the original Log-HS, with relatively little loss in classification accuracy, and is scalable to large datasets. Empirically, our framework compares very favorably with previous state of the art methods in terms of classification accuracy and especially in computational complexity.

As ongoing work, we are investigating the direction of replacing low-level hand-crafted features used in our current experiments with feature learning techniques, such as convolutional networks. Preliminary experiments on the Fish dataset show that we obtain an improvement of approximately 15% by using convolutional features in combination with the proposed approximate Log-HS distance formulation. We will report the full results in a future work.

References

[1] 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. 1, 2, 3

[2] R. Bhatia. *Positive Definite Matrices*. Princeton University Press, 2007. 2

[3] B. J. Boom, J. He, S. Palazzo, P. X. Huang, C. Beyan, H.-M. Chou, F.-P. Lin, C. Spampinato, and R. B. Fisher. A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. *Ecological Informatics*, 23:83–97, 2014. 6, 7

[4] P. Brodatz. *Textures: a photographic album for artists and designers*. Dover Publications, New York, 1966. 6, 7

[5] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *Tenth IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1597–1604, 2005. 6, 7

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2:27:1–27:27, 2011. 6

[7] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. Jensen-Bregman LogDet divergence with application to efficient similarity search for covariance matrices. *PAMI*, 35(9):2161–2174, 2013. 1, 6, 7, 8

[8] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1):1–34, 1999. 6, 7

[9] J. Dick, F. Kuo, and I. Sloan. High-dimensional integration: the quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013. 5

[10] A. Ess, B. Leibe, and L. V. Gool. Depth and appearance for mobile scene analysis. In *ICCV*, pages 1–8, 2007. 6, 7

[11] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 88(2). 6

[12] M. Faraki, M. Harandi, and F. Porikli. Approximate infinite-dimensional region covariance descriptors for image classification. In *ICASSP*, 2015. 2, 5, 6, 7, 8

[13] A. Feragen, F. Lauze, and S. Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *CVPR*, 2015. 2

[14] M. Harandi, M. Salzmann, and F. Porikli. Bregman divergences for infinite dimensional covariance matrices. In *CVPR*, 2014. 2, 5, 7, 8

[15] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on the Riemannian manifold of symmetric positive definite matrices. In *CVPR*, 2013. 1, 2, 3, 7

[16] G. Kylberg, M. Uppstroem, K.-O. Hedlund, G. Borgefors, and I.-M. Sintorn. Segmentation of virus particle candidates in transmission electron microscopy images. *Journal of Microscopy*, 245(2):140–147, 2012. 6

[17] G. Larotonda. Nonpositive curvature: A geometrical approach to Hilbert-Schmidt operators. *Differential Geometry and its Applications*, 25:679–700, 2007. 1, 3, 6

[18] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *CVPR*, pages 409–415, 2003. 6, 7

[19] P. Li, Q. Wang, W. Zuo, and L. Zhang. Log-Euclidean kernels for sparse representation and dictionary learning. In *ICCV*, pages 1601–1608, 2013. 1

[20] Z. Liao, J. Rock, Y. Wang, and D. Forsyth. Non-parametric filtering for geometric detail extraction and material representation. In *CVPR*, pages 963–970, 2013. 6, 7

[21] H. Minh, M. San Biagio, and V. Murino. Log-Hilbert-Schmidt metric between positive definite operators on Hilbert spaces. In *NIPS*, pages 388–396, 2014. 1, 2, 3, 4, 5, 6, 7

[22] H. Q. Minh. Affine-invariant Riemannian distance between infinite-dimensional covariance operators. In *Geometric Science of Information*, 2015. 3, 6

[23] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *IJCV*, 66(1):41–66, 2006. 2

[24] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on Lie algebra. In *CVPR*, volume 1, pages 728–735, 2006. 1

[25] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007. 2, 4

[26] M. Reed and B. Simon. *Methods of Modern Mathematical Physics: Fourier analysis, self-adjointness*. Academic Press, 1975. 4

[27] S. Si, C.-J. Hsieh, and I. Dhillon. Memory efficient kernel approximation. In *ICML*, pages 701–709, 2014. 2

[28] 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. 6

[29] 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. 8

[30] D. Tosato, M. Spera, M. Cristani, and V. Murino. Characterizing humans on Riemannian manifolds. *PAMI*, 35(8):1972–1984, Aug 2013. 1

[31] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. *PAMI*, 30(10):1713–1727, 2008. 1

[32] R. Wang, H. Guo, L. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *CVPR*, pages 2496–2503, 2012. 6, 7, 8

[33] A. Wendel and A. Pinz. Scene categorization from tiny images. In *31st Annual Workshop of the Austrian Association for Pattern Recognition*, pages 49–56, 2007. 6, 7, 8

[34] J. Yang, V. Sindhwani, H. Avron, and M. Mahoney. Quasi-Monte Carlo feature maps for shift-invariant kernels. In *ICML*, pages 485–493, 2014. 2, 4, 5

[35] S. K. Zhou and R. Chellappa. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel Hilbert space. *PAMI*, 28(6):917–929, 2006. 2