More About VLAD: A Leap from Euclidean to Riemannian Manifolds

Masoud Faraki Mehrtash T. Harandi Fatih Porikli
College of Engineering and Computer Science, Australian National University, Australia
NICTA*, Canberra Research Laboratory, Australia
{masoud.faraki,mehrtash.harandi,fatih.porikli}@nicta.com.au

Abstract

This paper takes a step forward in image and video coding by extending the well-known Vector of Locally Aggregated Descriptors (VLAD) onto an extensive space of curved Riemannian manifolds. We provide a comprehensive mathematical framework that formulates the aggregation problem of such manifold data into an elegant solution. In particular, we consider structured descriptors from visual data, namely Region Covariance Descriptors and linear subspaces that reside on the manifold of Symmetric Positive Definite matrices and the Grassmannian manifolds, respectively. Through rigorous experimental validation, we demonstrate the superior performance of this novel Riemannian VLAD descriptor on several visual classification tasks including video-based face recognition, dynamic scene recognition, and head pose classification.

1. Introduction

This paper extends the *Vector of Locally Aggregated Descriptors* (VLAD) [27] to general class of curved Riemannian manifolds, hence adding a novel dimension to the applicability of VLAD in tackling fundamental recognition problems in computer vision.

Our motivation stems from the fact that, in $\mathbb{R}^n$, coding local image or video descriptors using VLAD has been shown to be exceptionally successful in addressing a variety of challenging problems, such as image retrieval [27, 17], texture classification [9], and scene recognition [17]. The advantage of VLAD can be even more appreciated by noting that its high discriminatory power is achieved using rudimentary vector subtraction and addition operations, a negligible computational cost compared to more involved approaches like deep convolutional networks.

To put the discussion into perspective, describing images or videos by local descriptors is preferable to holistic representations when, for instance, the recognition problem pertains large intra-class variations, articulated shapes, self-occlusions, and changing backgrounds, to name a few. However, almost all previous studies [27, 41, 50, 17, 9] that extracted local descriptors in the form of vectors disregarded the underlying intrinsic manifold structures, probably because proven and general techniques to aggregate non-vectorized structures are scarce.

On a related note, structured representations such as Region Covariance Descriptors (RCovD) and linear subspaces have been shown to provide robust and efficient representations for a wide range of tasks [22, 11, 51, 12, 46, 48, 49]. Therefore, a curious mind might inquire

- can we extract structured descriptors from visual data and then aggregate them in a fashion similar to VLAD to obtain more discriminating representations?
- is there any mathematical framework that helps us formulate the aggregation problem into an elegant yet general and accurate solution?

This paper provides answers to the aforementioned questions by introducing the Riemannian version of the conventional VLAD, called R-VLAD, a new coding approach that enables fusing local descriptors on curved spaces.

More specifically, we present a universal framework for constructing a rich representation out of local image or video descriptors where each local descriptor is indeed a point on a Riemannian manifold. The conventional VLAD algorithm can hence be considered as a special case of our Riemannian formulation when the manifold is chosen to be the Euclidean space.

We then turn our attention to two widely utilized Riemannian manifolds in vision, namely the manifold of Symmetric Positive Definite matrices (SPD) and the manifold of linear subspaces, known as the Grassmannian manifold. To this end, we develop the sister family of R-VLAD by exploiting other forms of metrics defined on SPD matrices and Grassmannian manifolds. In particular, we make use of the Stein [43] and Jeffrey [52] divergences on SPD matrices and

*NICTA is funded by the Australian Government through the Department of Communications, as well as the Australian Research Council through the ICT Centre of Excellence program.

the projection distance [20] on Grassmannian manifolds to yield computationally more efficient versions of R-VLAD.

Our experiments demonstrate the superiority of the proposed R-VLAD descriptor against several baseline and state-of-the-art methods such as the Weighted ARray of COvariances (WARCO) of Tosato *et al*. [46] for head pose classification, the Adaptive Deep Network Template (ADNT) of Hayat *et al*. [23] for video-based face classification, and the Bag of Spatiotemporal Energy (BoSE) of Feichtenhofer *et al*. [13] for dynamic scene recognition .

To the best of our knowledge, using the standard protocol, the proposed R-VLAD achieves top results on standard benchmarks: 85% for HOCoffe [46], 79.1% for HDM05 [32], 79.9% for YouTube Celebrities [28], 97.6% for Dyntex++ [14], and 99.8% for UPENN [10].

2. Related Work

Since the late nineties, *Bag of Words* (BoW) [40, 18, 30, 42] and its extensions [29, 19] have been the de facto driving force for image and video representation. Two notable examples are Video Google [42] for object matching in videos and Spatio-temporal Pyramid Matching (SPM) [29] for scene classification.

Broadly speaking, the local models such as BoW, *Fisher Vectors* (FV) [37], and VLAD [27] can benefit from powerful local feature descriptors (*e.g*. SIFT [31]), which to some extent provide robustness to transformations such as scaling, translation, and occlusion. Furthermore, the resulting vector can be compared using the Euclidean distance norms and utilized [36] in conventional classifiers (*e.g*. Support Vector Machines (SVM)).

The VLAD descriptor, our main focus in this work, is a coding scheme similar in spirit to the earlier FV [37]. It is able to provide compact codes by capturing certain aspects of the distribution of features. While inheriting the useful properties of BoW, VLAD departs from it as it encodes the differences from the cluster centers rather than counting the number of assignments to them.

Several recent studies [26, 7, 2] suggest that promising performances on various benchmarks can be attained by effective use of the original VLAD descriptor. For example, Gong *et al*. proposed to exploit VLAD to pool the activations of Deep Convolutional Neural Networks. Impressively, the results of multi-scale VLAD reported in [17] outperformed various state-of-the-art methods. One example is the scene classification on the MIT dataset, where the multi-scale VLAD with only 4096 features comfortably outperformed the mixture of FV and bag-of-parts, which used 221,550 features for its decision [17]. Another example is the work of Cimpoi *et al*. who showed that the VLAD descriptor is preferable to others for the task of texture recognition [9]. Some other recent advances include better normalization schemes for the VLAD descriptor [2], supervised codebook learning for VLAD [34], VLAD with higher order statistics [34], and VLAD for action recognition [25].

3. Riemannian VLAD

In this section, we derive a general formulation for Riemannian VLAD. In doing so, we start by studying the conventional VLAD formulation, and in particular through its predecessor, the Fisher Vector (FV) [37].

3.1. Conventional VLAD in Euclidean Spaces

Given a set of local descriptors $\mathcal{X} = \{\boldsymbol{x}_t\}_{t=1}^m, \boldsymbol{x}_t \in \mathbb{R}^d$ extracted from an image or video, let us assume that $\mathcal{X}$ admits a probability density function in the form of a Gaussian Mixture Model (GMM)

$$p(\boldsymbol{x}_t|\lambda) = \sum_{i=1}^{K} \omega_i \mathcal{N}(\boldsymbol{x}_t|\, \boldsymbol{\mu}_i, \Sigma_i) \ ,$$

with $\lambda = \{\omega_i, \boldsymbol{\mu}_i, \Sigma_i\}$ being the mixture weight, mean and covariance of the Gaussian components, respectively.

In the FV method, $\mathcal{X}$ is encoded through its *score function*[1]. Related to VLAD is the gradient part with respect to $\boldsymbol{\mu}_i$ which has the following form

$$\nabla_{\mu_i} \log p(\mathcal{X}|\lambda) = \sum_{t=1}^{m} \gamma_i(\boldsymbol{x}_t) \Sigma_i^{-1} \left(\boldsymbol{\mu}_i - \boldsymbol{x}_t\right) \ , \quad (1)$$

where $\gamma_i(\boldsymbol{x}_t)$ is the soft-assignment of $\boldsymbol{x}_t$ to the i-th Gaussian component, *i.e*.,

$$\gamma_i(\boldsymbol{x}_t) = \frac{\omega_i \mathcal{N}(\boldsymbol{x}_t|\, \boldsymbol{\mu}_i, \Sigma_i)}{\sum_{j=1}^{K} \omega_j \mathcal{N}(\boldsymbol{x}_t|\, \boldsymbol{\mu}_j, \Sigma_j)} \ .$$

In VLAD, the input space $\mathbb{R}^d$ is partitioned into K Voronoi cells by means of a codebook $\mathcal{C}$ with centers $\{\boldsymbol{c}_i\}_{i=1}^K$, $\boldsymbol{c}_i \in \mathbb{R}^d$. For the aforementioned query set $\mathcal{X}$, the VLAD code $V \in \mathbb{R}^{Kd}$ is obtained by concatenating K Local Difference Vectors (LDV) $\boldsymbol{v}_i$ storing the differences $\boldsymbol{c}_i - \boldsymbol{x}_t$ in each cell, *i.e*.,

$$\boldsymbol{v}_i = \sum_{\boldsymbol{x}_t \in \boldsymbol{c}_i} \boldsymbol{c}_i - \boldsymbol{x}_t \ , \quad (2)$$

where $\boldsymbol{x} \in \boldsymbol{c}_i$ means that the local descriptor $\boldsymbol{x}$ belongs to the Voronoi defined by $\boldsymbol{c}_i$, *i.e*., the closest codeword to $\boldsymbol{x}$ is $\boldsymbol{c}_i$.

Direct comparison between Eq. (1) and Eq. (2) reveals the following about VLAD

1. In contrast to FV which uses soft assignment of local descriptors to the Gaussians, VLAD exploits a hard assignment scheme.

2. Unlike FV, the covariance matrices of the mixture components with VLAD are assumed to be diagonal and fixed, *i.e*., $\Sigma_i = \sigma \mathbf{I}_d, \ \forall i \in \{1, 2, \cdots, K\}$.

[1] In statistics, the score function is the gradient of the log-likelihood of the data on the model.

The take-home message here is that the LDV is indeed the gradient of the Euclidean distance[2] and the VLAD framework is a clever simplification of the FV algorithm. With this introduction, we are now ready to introduce Riemannian VLAD (R-VLAD).

3.2. Extension to Riemannian Manifolds

Now let us assume that $\mathcal{X} = \{\boldsymbol{x}_t\}_{t=1}^{m}$, $\boldsymbol{x}_t \in \mathcal{M}$ and $\mathcal{C} = \{\boldsymbol{c}_k\}_{k=1}^{K}$, $\boldsymbol{c}_k \in \mathcal{M}$, are a set of local descriptors (extracted from a query image or video) and codewords on a Riemannian manifold $\mathcal{M}$, respectively. The R-VLAD descriptor on $\mathcal{M}$ is obtained once we have the followings tools at our disposal

- a metric $\delta(\boldsymbol{x}, \boldsymbol{y}) : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ required to determine how the local descriptors should be assigned to the codewords.
- operators to perform the role of vector addition or subtraction on $\mathcal{M}$.

Since a Riemannian manifold is a metric space, one could seamlessly use the geodesic distance $\delta_g : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ to address the first requirement. As for the second requirement, we note that on a Riemannian manifold, one can see a vector $\overrightarrow{ab}$ (attached at point $\boldsymbol{a}$) as a vector of the tangent space at $\boldsymbol{a}$, *i.e.* $T_{\boldsymbol{a}}\mathcal{M}$. Therefore, subtraction on a Riemannian manifold can be attained through the logarithm map, $\log_{\boldsymbol{a}}(\cdot) : \mathcal{M} \to T_{\boldsymbol{a}}\mathcal{M}$[3]. This concept has been used widely in the literature. For example, vector subtraction through the logarithm map was used to address the problem of interpolation and filtering [35], sparse coding [24], and dimensionality reduction [15], to name a few.

The aforementioned discussion hints towards devising the R-VLAD as follows

- exploit the geodesic distance to determine the closest local descriptors to each codeword.
- build a Riemannian LDV per codeword using the tangent space attached to each codeword on the manifold.

Since the pole of the tangent space, *i.e.*, $\boldsymbol{c}_i$ is fixed, the outputs of the logarithm map are compatible with each other and no further special care (*e.g.*, parallel transport) is required[4]. Therefore, Eq. (2) on a curved Riemannian manifold boils down to

$$\boldsymbol{v}_i = \sum_{\boldsymbol{x}_t \in \boldsymbol{c}_i} \log_{\boldsymbol{c}_i}(\boldsymbol{x}_t) \ . \tag{3}$$

[2]To be more precise, the gradient of a Gaussian function where the associated normalization terms are discarded.

[3]Due to the lack of space, we skip rigorous definition of tangent vectors, logarithm map and other concepts of Riemannian geometry. The interested reader is referred to textbooks on Riemannian geometry for a formal treatment.

[4]To be precise, this argument is valid as long as $\boldsymbol{x}_t$ is not in the cut locus of $\boldsymbol{c}_i$. This is of course not a very restricting assumption as in many manifolds (*e.g.*, the SPD manifold) the cut locus is indeed empty.

While being perfectly valid, for reasons that become clear later, we are interested in having a more general formulation for R-VLAD. More specifically, rather than having a method that only works with the geodesic distance, we would like to extend our formulation such that any metric on $\mathcal{M}$ can be used.

Obviously, for a new metric $\delta : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$, we need to only take care of the second requirement. Since the LDV can be understood as the gradient of the distance function in the Euclidean case (see § 3.1), it is tempting to define the LDV on $\mathcal{M}$ as $\nabla_{\boldsymbol{c}_i}\delta^2(\boldsymbol{c}_i, \boldsymbol{x}_t)$[5]. The following theorem reinforces this idea even more.

Theorem 1. *For a Riemannian manifold $\mathcal{M}$, the gradient of the geodesic distance function, $\delta_g : \mathcal{M} \times \mathcal{M} \to \mathbb{R}^+$ is given by*

$$\nabla_{\boldsymbol{x}}\delta_g^2(\boldsymbol{x}, \boldsymbol{y}) = -2\log_{\boldsymbol{x}}(\boldsymbol{y}). \tag{4}$$

Proof. The interested reader is referred to [45] for the proof of this theorem. □

Unfortunately, choosing $\nabla_{\boldsymbol{c}_i}\delta^2(\boldsymbol{c}_i, \boldsymbol{x}_t)$ for LDV will not work in practice. The main reason being that for δ_g, the norm of $\nabla_{\boldsymbol{x}}\delta_g^2(\boldsymbol{x}, \boldsymbol{y})$ is related directly to the metric, *i.e.*,

$$\|\nabla_{\boldsymbol{x}}\delta_g^2(\boldsymbol{x}, \boldsymbol{y})\|^2 = 4\|\log_{\boldsymbol{x}}(\boldsymbol{y})\|^2 = 4\delta_g^2(\boldsymbol{x}, \boldsymbol{y}).$$

This is of course inherited to the Euclidean space when the metric is chosen to be the geodesic distance, *i.e.*, the Euclidean distance. However, this will not generalize to other metrics as shown by the following example.

Example 1. *Fig. 1 shows the behavior of $\nabla_{\boldsymbol{X}}\delta^2(\boldsymbol{X}, \boldsymbol{Y})$ by varying $\delta^2(\boldsymbol{X}, \boldsymbol{Y})$ for the projection metric on the Grassmann manifold $\mathcal{G}_3^2$ (see § 5 for the equations). Interestingly, the norm of the gradient will start decreasing while point $\boldsymbol{Y}$ gets farther away from $\boldsymbol{X}$. This means, during encoding, a point which should contribute significantly to the output, can act as an insignificant point, hence deteriorating the discriminatory power of the descriptor.*

The aforementioned example provides us with the following guideline for constructing an LDV on $\mathcal{M}$.

- the length of the LDV should represent the metric considered on $\mathcal{M}$.

As such, we propose the following form of LDV for our general R-VLAD descriptor (see Algorithm 1 for a step-by-step on the R-VLAD technique).

$$\boldsymbol{v}_i = \sum_{\boldsymbol{x}_t \in \boldsymbol{c}_i} \delta(\boldsymbol{c}_i, \boldsymbol{x}_t) \frac{\nabla_{\boldsymbol{c}_i}\delta^2(\boldsymbol{c}_i, \boldsymbol{x}_t)}{\|\nabla_{\boldsymbol{c}_i}\delta^2(\boldsymbol{c}_i, \boldsymbol{x}_t)\|} \ . \tag{5}$$

[5]On an abstract Riemannian manifold $\mathcal{M}$, the gradient of a smooth real function f at a point $x \in \mathcal{M}$, denoted by $\nabla_x f$, is the element of $T_x\mathcal{M}$ satisfying $\langle \text{grad} f(x), \zeta \rangle_x = Df_x[\zeta]$ for all $\zeta \in T_x\mathcal{M}$, where $Df_x[\zeta]$ denotes the directional derivative of f at x in the direction of ζ.

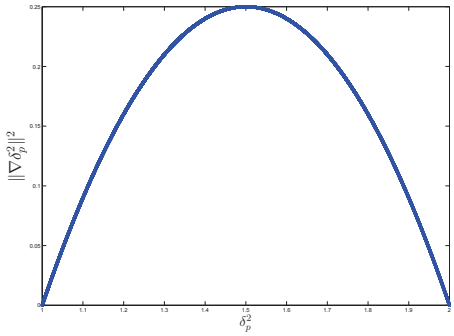

Figure 1: Illustration of the squared norm of the gradients vs distance for the projection distance on $\mathcal{G}_3^2$.

Algorithm 1 The proposed R-VLAD algorithm

Input:

- local descriptors $\mathcal{X} = \{\boldsymbol{x}_t\}_{t=1}^m, \boldsymbol{x}_t \in \mathcal{M}$, extracted from a query image or video,
- codebook $\mathcal{C} = \{\boldsymbol{c}_k\}_{k=1}^K, \boldsymbol{c}_k \in \mathcal{M}$

Output:

- $\boldsymbol{V}(\mathcal{X})$ the Riemannian VLAD representation of $\mathcal{X}$

1: **for** $i = 1 \rightarrow k$ **do**
2: Find $\boldsymbol{x}_t \in \boldsymbol{c}_i$, all nearest query points from $\mathcal{X}$ to $\boldsymbol{c}_i$
3: Compute $\boldsymbol{v}_i$, i-th Local Difference Vector (LDV), using Eq. (5)
4: **end for**
5: Concatenate the resulting LDVs to form the final descriptor, *i.e.*, $\boldsymbol{V}(\mathcal{X}) = \left[\boldsymbol{v}_1^T, \boldsymbol{v}_2^T, \cdots, \boldsymbol{v}_k^T\right]^T$

Remark 1. *In line with the recommendations in [38], post-processing of VLAD codes could increase the discriminatory power of the codes. In practice, we normalize the R-VLAD codes in two steps. First, an element-wise power normalization is performed using the transfer function* $y : \mathbb{R} \rightarrow \mathbb{R},\ y(x) = sign(x)\sqrt{|x|}$, *where* x *is the element of VLAD vector and* $|\cdot|$ *denotes absolute value. This is to avoid having a concentrated distribution around zero. The power normalization is followed by an* ℓ_2 *normalization to make the energy of descriptors uniform.*

In the following sections, we develop the R-VLAD for two widely used manifolds in vision, *i.e.*, the SPD and the Grassmannian manifolds (see Table 1 for a quick peak at the studied metrics and the associated gradients as required by Eq. (5)). Before concluding this section and for the sake of completeness, we discuss how a Riemannian codebook can be learned from training samples.

3.3. k-Means on Riemannian Manifolds

Given an abstract manifold $\mathcal{M}$ with an associated metric $\delta : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+$, we train a codebook similar to standard k-means using an EM-based approach. The algorithm starts by selecting k points from the training data randomly as the cluster centers. In the E-step, each of the points are assigned to the nearest cluster center using δ. Then in the M-step, the cluster centers are re-computed using the Fréchet mean.

Definition 1. *The Fréchet mean for a set of points* $\{\boldsymbol{x}_i\}_{i=1}^m$, $\boldsymbol{x}_i \in \mathcal{M}$ *is local minimizer of the cost function*

$$c^* = \arg\min_c \sum_{i=1}^m \delta^2(c, \boldsymbol{x}_i) \ . \tag{6}$$

In general, an analytic solution for Eq. (6) cannot be sought and iterative schemes that exploit the logarithm and exponential maps must be employed [35]. This, for a high-dimensional manifold with a big m, could easily become overwhelming. One reason in generalizing the R-VLAD to work with an arbitrary metric comes from the fact that, unlike the general case, for some metrics Eq. (6) has an analytic solution. We will provide more details on how Eq. (6) can be solved for the special cases of interest in this paper later.

4. R-VLAD on SPD Manifold

The space of $d \times d$ SPD matrices is mostly studied when endowed with a Riemannian metric and thus forming a Riemannian manifold [35]. The Affine Invariant Riemannian Metric (AIRM) is probably the most popular Riemannian structure for analyzing SPD matrices [35].

Definition 2. *The geodesic distance* $\delta_g : \mathcal{S}_{++}^d \times \mathcal{S}_{++}^d \rightarrow [0, \infty)$ *induced by the AIRM is defined as*

$$\delta_g(\boldsymbol{X}, \boldsymbol{Y}) = \| \log(\boldsymbol{X}^{-1/2} \boldsymbol{Y} \boldsymbol{X}^{-1/2}) \|_F \ , \tag{7}$$

where $\log(\cdot)$ *is the matrix principal logarithm.*

Beside the AIRM, two types of Bregman divergences, namely the Jeffrey [52] and Stein [43] are widely used to measure similarities on SPD manifolds.

Definition 3. *The J divergence (also known as Jeffrey or symmetric KL divergence)* $\delta_J : \mathcal{S}_{++}^d \times \mathcal{S}_{++}^d \rightarrow [0, \infty)$ *is a symmetric type of Bregman divergence and is defined as*

$$\delta_J^2(\boldsymbol{X}, \boldsymbol{Y}) \triangleq \frac{1}{2} \operatorname{Tr}(\boldsymbol{X}^{-1}\boldsymbol{Y}) + \frac{1}{2} \operatorname{Tr}(\boldsymbol{Y}^{-1}\boldsymbol{X}) - d \ . \tag{8}$$

Definition 4. *The Stein metric* $\delta_S : \mathcal{S}_{++}^d \times \mathcal{S}_{++}^d \rightarrow [0, \infty)$ *is also a symmetric type of Bregman divergence and is defined as*

$$\delta_S^2(\boldsymbol{X}, \boldsymbol{Y}) \triangleq \ln \det \left(\frac{\boldsymbol{X} + \boldsymbol{Y}}{2} \right) - \frac{1}{2} \ln \det(\boldsymbol{X}\boldsymbol{Y}) \ . \tag{9}$$

As for the δ_g, the Fréchet mean is obtained using an iterative approach [35]. The same holds for the Stein metric as a result of the following theorem.

Theorem 2. *The Fréchet mean of a set of SPD matrices* $\{\boldsymbol{X}_i\}_{i=1}^m \in \mathcal{S}_{++}^d$ *with* δ_S *is obtained iteratively via*

$$\boldsymbol{\mu}^{(t+1)} = \left[\frac{1}{m} \sum_{i=1}^m \left(\frac{\boldsymbol{X}_i + \boldsymbol{\mu}^{(t)}}{2} \right)^{-1} \right]^{-1} . \tag{10}$$

Table 1: Metrics and associated gradients on the SPD and Grassmannian manifold.

Manifold	Metric	$\delta^2(\boldsymbol{X},\boldsymbol{Y})$	$\nabla_{\boldsymbol{X}}\delta^2$
$\mathcal{S}_{++}^d$	geodesic	$\|\log(\boldsymbol{X}^{-1/2}\boldsymbol{Y}\boldsymbol{X}^{-1/2})\|_F^2$	$2\boldsymbol{X}^{1/2}\log(\boldsymbol{X}^{-1/2}\boldsymbol{Y}\boldsymbol{X}^{-1/2})\boldsymbol{X}^{1/2}$
$\mathcal{S}_{++}^d$	Stein	$\ln\det\left(\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right)-\frac{1}{2}\ln\det(\boldsymbol{X}\boldsymbol{Y})$	$\boldsymbol{X}(\boldsymbol{X}+\boldsymbol{Y})^{-1}\boldsymbol{X}-\frac{1}{2}\boldsymbol{X}$
$\mathcal{S}_{++}^d$	Jeffrey	$\frac{1}{2}\operatorname{Tr}(\boldsymbol{X}^{-1}\boldsymbol{Y})+\frac{1}{2}\operatorname{Tr}(\boldsymbol{Y}^{-1}\boldsymbol{X})-d$	$\frac{1}{2}\boldsymbol{X}\left(\boldsymbol{Y}^{-1}-\boldsymbol{X}^{-1}\boldsymbol{Y}\boldsymbol{X}^{-1}\right)\boldsymbol{X}$
$\mathcal{G}_d^p$	geodesic	$\|\Theta\|^2$	No analytic form
$\mathcal{G}_d^p$	projection	$2p-2\|\boldsymbol{X}^T\boldsymbol{Y}\|_F^2$	$-4\left(\mathbf{I}_d-\boldsymbol{X}\boldsymbol{X}^T\right)\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{X}$

Proof. See [8] for the proof. □

However, with the Jeffrey divergence, we have the luxury of obtaining the Fréchet mean analytically.

Theorem 3. *The Fréchet mean of a set of SPD matrices* $\{\boldsymbol{X}_i\}_{i=1}^m \in \mathcal{S}_{++}^d$ *with* δ_J *is*

$$\boldsymbol{\mu} = \boldsymbol{P}^{-1/2}(\boldsymbol{P}^{1/2}\boldsymbol{Q}\boldsymbol{P}^{1/2})^{1/2}\boldsymbol{P}^{-1/2}\ , \tag{11}$$

where $\boldsymbol{P}=\sum_i \boldsymbol{X}_i^{-1}$ *and* $\boldsymbol{Q}=\sum_i \boldsymbol{X}_i$.

Proof. The solution is obtained by zeroing out the derivative of $\sum_i^m \delta_J^2(\boldsymbol{X}_i,\boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$. The proof is relegated to the supplementary material. □

The gradient of a function $f:\mathcal{S}_{++}^d\to\mathbb{R}$ at $\boldsymbol{X}$ has the following form on $\mathcal{S}_{++}^d$ [44]

$$\nabla_{\boldsymbol{X}} f = \boldsymbol{X}\operatorname{sym}(Df)\boldsymbol{X}, \tag{12}$$

where $\operatorname{sym}(\boldsymbol{X})=0.5(\boldsymbol{X}+\boldsymbol{X}^T)$ and Df is the derivative of the function $f:\mathbb{R}^{d\times d}\to\mathbb{R}$ with respect to $\boldsymbol{X}$.

The derivatives of $D\delta_S^2$ and $D\delta_J^2$ are reported in [8][6]. From [8] we can deduce the gradients required in the R-VLAD algorithm as depicted in Table 1.

Computational Cost

The computational loads of computing δ_g^2, δ_J^2 and δ_S^2 are $4d^3$, $8/3d^3$ and d^3, respectively [8]. Computing the gradient of δ_g^2 requires an eigenvalue decomposition (for computing principal matrix logarithm) which adds up to a total of $9d^3$ flops for δ_g^2 (considering the matrix multiplications). For δ_J^2 and δ_S^2, computing gradient just requires a matrix inversion which is $O(d^3)$. As such, the computational load of R-VLAD using δ_J^2 and δ_S^2 is $O(17/3d^3)$ and $O(4d^3)$, respectively.

5. R-VLAD on Grassmannian

The space of p-dimensional linear subspaces of $\mathbb{R}^d$ for $0<p<d$ is not a Euclidean space, but a Riemannian manifold known as the Grassmannian $\mathcal{G}_p^d$ [1]. A point on the Grassmann manifold $\mathcal{G}_d^p$ may be represented by an arbitrary $d\times p$ matrix with orthogonal columns, *i.e.*, $\boldsymbol{X}\in\mathcal{G}_p^d \Rightarrow \boldsymbol{X}^T\boldsymbol{X}=\mathbf{I}_p$[7]. For the Grassmannian, the geodesic distance between two points $\boldsymbol{X}$ and $\boldsymbol{Y}$ is given by

$$\delta_g(\boldsymbol{X},\boldsymbol{Y}) = \|\Theta\|_2\ , \tag{13}$$

where Θ is the vector of principal angles between $\boldsymbol{X}$ and $\boldsymbol{Y}$ [1].

In addition to the geodesic distance, a popular metric on $\mathcal{G}_d^p$ is the projection metric $\delta_P:\mathcal{G}_d^p\times\mathcal{G}_d^p\to\mathbb{R}^+$ defined as [22, 20]

$$\delta_P^2(\boldsymbol{X},\boldsymbol{Y}) = \|\boldsymbol{X}\boldsymbol{X}^T-\boldsymbol{Y}\boldsymbol{Y}^T\|_F^2, \tag{14}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Unlike δ_g which does not have an analytic form for the Fréchet mean (see [47] for more details), the projection metric has the following interesting property.

Theorem 4 (Closed-Form Mean)**.** *The Fréchet mean for a set of points* $\{\boldsymbol{X}_i\}_{i=1}^m$, $\boldsymbol{X}_i\in\mathcal{G}_d^p$ *based on* δ_P *admits a closed-form solution. That is the* p *largest eigenvectors of* $\sum_{i=1}^m \boldsymbol{X}_i\boldsymbol{X}_i^T$.

Proof. The solution is obtained by maximizing $\operatorname{Tr}\{\boldsymbol{\mu}^T\left(\sum_{i=1}^m \boldsymbol{X}_i\boldsymbol{X}_i^T\right)\boldsymbol{\mu}\}$ and considering the orthogonality constraint $\boldsymbol{\mu}^T\boldsymbol{\mu}=\mathbf{I}_p$. The proof is relegated to the supplementary material due to the lack of space. □

The gradient of a function on Grassmannian, *i.e.*, $f:\mathcal{G}_d^p\to\mathbb{R}$ has the form

$$\nabla_{\boldsymbol{X}} f = \left(\mathbf{I}_d-\boldsymbol{X}\boldsymbol{X}^T\right)Df, \tag{15}$$

where Df is a $d\times p$ matrix of partial derivatives of f with respect to the elements of $\boldsymbol{X}$, *i.e.*,

$$Df_{i,j} = \frac{\partial f}{\partial X_{i,j}}.$$

The logarithm map (and also the exponential map) on Grassmannian does not have an analytic form. However, numerical methods for computing both mappings do exist. In particular, we will use the formulation introduced in [5] to compute R-VLAD using the geodesic distance.

[6]Note that in Table 3 of [8] a scalar factor of 0.5 is wrongly dropped from the Jeffrey divergence (KLDM according to [8]). Also please note that the gradient reported in [8] is the Euclidean gradient not the Riemannian as required here.

[7]A point on the Grassmannian $\mathcal{G}_d^p$ is a subspace spanned by the columns of a $d\times p$ full rank matrix and should therefore be denoted by $\operatorname{span}(\boldsymbol{X})$. With a slight abuse of notation, here we call $\boldsymbol{X}$ a Grassmannian point whenever it represents a basis for a subspace.

As for the projection metric, using Eq. (15) and noting that $\delta_P^2(\boldsymbol{X},\boldsymbol{Y}) = 2p - 2\|\boldsymbol{X}^T\boldsymbol{Y}\|_F^2$ leads to the following analytic form for the gradient as required in Eq. (5)

$$\nabla_{\boldsymbol{X}}\delta_P^2(\boldsymbol{X},\boldsymbol{Y}) = -4\left(\mathbf{I}_d - \boldsymbol{X}\boldsymbol{X}^T\right)\boldsymbol{Y}\boldsymbol{Y}^T\boldsymbol{X}. \qquad (16)$$

Computational Cost

The computational load of coding in R-VLAD is dictated by the complexity of the used metric δ^2 and its gradient. On top of this, one should pay attention to the complexity of Riemannian k-means. As long as the complexity of coding is considered, we note that δ_g^2 on Grassmannian is obtained through SVD decomposition. As such, computing δ_g^2 requires dp^2+p^3 flops on $\mathcal{G}_d^p$. In contrast, the complexity of computing δ_P^2 on $\mathcal{G}_d^p$ is dp^2.

Computing the gradient of δ_g^2 (or logarithm map) using a very efficient implementation requires a matrix inversion of size $p \times p$, two matrix multiplications of size $d \times p$, and a thin SVD of size $d \times p$. Computing thin SVD using a stable algorithm like the Golub-Reinsch [16] requires $14dp^2+8p^3$ flops. This adds up to a total of $O\big(10p^3 + 17dp^2\big)$ flops for one local descriptor. As for δ_p^2, computing the gradient according to the Table 1 demands for $4dp^2$ operations. This results in a total of $5dp^2$ flops for the projection metric.

To give the reader a better sense on the computational complexity of R-VLAD using δ_g^2 and δ_P^2, we measured the coding time for 1000 videos each with its own set of local descriptors on $\mathcal{G}_{177}^6$ (this is an example of the Grassmannian we will use in our experiments later). On a quad-core machine using Matlab, coding time for δ_P^2 and δ_g^2 were observed to be around 155 and 440 seconds, respectively.

6. Experiments

This section presents comparative evaluation results of our proposal against the baseline and state-of-the-art for a number of visual recognition problems defined on the SPD and Grassmannian manifold. In all our experiments, a set of overlapping blocks or cubes are extracted from images or videos. Each block or cube is then represented by an RCovD or a linear subspace, hence it corresponds to a point on the SPD or the Grassmannian manifold, respectively. Different algorithms tested in our experiments are

BoW$_G$: Riemannian BoW model using geodesic distance. An image or video is described by the histogram of its building blocks or cubes using geodesic distance. The codebook is learned by Riemannian k-means algorithm as described in § 3.3.

BoW$_{LE}$: BoW model trained by flattening the manifold through a fixed tangent space. We follow the terminology introduced in [3] and label this as Log-Euclidean BoW or for short **BoW**$_{LE}$.

VLAD$_{LE}$: Similar in concept to **BoW**$_{LE}$ but instead of BoW, we assess the performance of VLAD by flattening the manifold through a fixed tangent space.

R-VLAD$_G$: R-VLAD using geodesic distance.

R-VLAD$_{J/S/P}$: R-VLAD using the Jeffrey, Stein, or projection metrics.

Besides the Log-Euclidean and **BoW**$_G$ methods that serve as baseline methods, we will exclusively consider previous state-of-the-art algorithms for each studied problem to demonstrate the power of R-VLAD.

For classification, the VLAD descriptors, either Log-Euclidean or R-VLAD, are fed to a Nearest Neighbor (NN) or a linear Support Vector Machine (SVM) [6]. We separately report the performances for these two classifiers. In all the experiments, the size of codebook for VLAD is set to 16. This value is obtained empirically as a good compromise between computational cost and classification accuracy. For the **BoW**$_G$, we will report the best accuracy by searching over various codebook sizes.

6.1. SPD Manifold

For tests on the SPD manifold, an image or video is described by a set of RCovDs. More specifically, given a block $I(x,y)$[8] of size $W \times H$, let $\mathbb{O} = \{\boldsymbol{o}_i\}_{i=1}^r$, $\boldsymbol{o}_i \in \mathbb{R}^d$ be a set of r observations extracted from $I(x,y)$, *e.g.*, $\boldsymbol{o}_i$ concatenates intensity values, gradients, filter responses, etc. for image pixel i. Then, block I can be represented by the $d \times d$ RCovD using

$$\boldsymbol{C}_I = \frac{1}{r-1}\sum_{i=1}^{r}(\boldsymbol{o}_i - \overline{\boldsymbol{o}})(\boldsymbol{o}_i - \overline{\boldsymbol{o}})^T \ , \qquad (17)$$

where $\overline{\boldsymbol{o}} = \frac{1}{r}\sum_{i=1}^r \boldsymbol{o}_i$.

Head Pose Classification:

As our first experiment, we study the problem of head pose (orientation) classification. To this end, we utilized the Heads Of Coffee break (HOCoffee) dataset [46] which presents outdoor images captured by a head detector for the purpose of automatically detecting social interactions. This dataset is composed of 18,117 low-resolution images of size 50×50 pixels. The dataset comes with a predefined test protocol in which 9522 images are considered as training data and the remaining 8595 images are used for evaluation.

Similar to [46], we used a Difference Of Offset Gaussian (DOOG) filter-bank along color and image gradients to describe each image. More specifically, the feature vector assigned to each pixel in the image is

$$\boldsymbol{o}_{x,y} = \Big[I_L(x,y), I_a(x,y), I_b(x,y), \sqrt{I_x^2+I_y^2}, \\ arctan\Big(\frac{|I_x|}{|I_y|}\Big), G_1(x,y), G_2(x,y), \cdots, G_8(x,y)\Big] \ ,$$

[8] A similar discussion holds for cubes extracted from videos.

Table 2: Recognition accuracies for the HOCoffee [46] and HDM05 [32] datasets.

Method	HOCoffee	HDM05
Previous Best	80.8 [46]	73.3 ± 11.4 [21]
BOW$_G$-NN	81.8	65.3 ± 14.1
BOW$_G$-SVM	80.0	71.6 ± 7.7
BOW$_{LE}$-NN	81.6	50.4 ± 11.2
BOW$_{LE}$-SVM	78.9	65.9 ± 8.5
VLAD$_{LE}$-NN	82.4	65.4 ± 13.5
VLAD$_{LE}$-SVM	82.4	71.2 ± 10.6
R-VLAD$_G$-NN	**85.0**	78.1 ± 5.8
R-VLAD$_G$-SVM	84.3	**79.1 ± 7.5**
R-VLAD$_S$-NN	84.9	72.5 ± 10.1
R-VLAD$_S$-SVM	84.2	74.1 ± 5.2
R-VLAD$_J$-NN	84.5	74.3 ± 8.2
R-VLAD$_J$-SVM	83.3	76.5 ± 11.6

where $I_c(x, y)$, $c \in \{L, a, b\}$, denotes the CIELab color information at position (x, y), I_x and I_y are luminance derivatives, and $G_i(x, y)$ denotes the response of the i-th DOOG centered at $I_L(x, y)$. Therefore, each local covariance descriptor is on $\mathcal{S}^{13}_{++}$.

In the second column of Table 2, we report the recognition accuracies of all the studied methods for this dataset. Several conclusion can be drawn here. First of all, the local approach, even the simple **BOW**$_G$-NN, outperforms the previous state-of-the-art method. The R-VLAD technique with all studied metrics achieve the best performances (with a NN classifer), with **R-VLAD**$_G$ being the overall winner in terms of accuracy. However, the performance of R-VLAD with the Stein and Jeffrey is on par or slightly worse than that of the geodesic solution while being at least 27 times faster in coding and 65 times faster (especially for the case of Jeffrey) in the training phase. We also observe that the proposed R-VLAD method is significantly superior as compared to the Log-Euclidean methods, which suggests that the underlying Riemannian structure is better exploited in R-VLAD.

Action Recognition from Motion Capture Data:
As our second experiment, we tackled the task of human action recognition from the skeletal information. To this end, we used HDM05 dataset [32], which contains 14 different human actions performed by 5 subjects.

We followed the protocol of [21] in which only the real-world location of 4 joints related to arms and legs are utilized for generating RCovDs. Therefore, RCovDs of size 12×12, extracted with a temporal overlap of 75% were used as local descriptors for an action. We used a leave-one-subject-out protocol, where the data of 4 subjects are used for training and the remaining one is used for testing.

The average recognition accuracies along the standard deviations are reported in the last column of Table 2. From this table we conclude that the R-VLAD equipped with a linear SVM classifier outperforms the state-of-the-art [21] regardless of its metric. The maximum accuracy is obtained by **R-VLAD**$_G$ which is nearly 6 percentage points better than [21]. Furthermore, the **R-VLAD**$_G$ is significantly better than **R-VLAD**$_S$ and **R-VLAD**$_J$ which makes it the preferred technique if coding time is not an issue.

Moreover, we evaluated the performance of the VLAD in Euclidean space (**VLAD**$_E$) using very small to large codebook sizes to obtain signatures with the dimensionality similar or greater than that of R-VLAD's signatures. We observed that R-VLAD is significantly superior to **VLAD**$_E$. For instance, the best accuracy of **VLAD**$_E$ on the HOCoffee and HDM05 datasets are 79.9% and 69.4%, respectively.

6.2. Grassmannian Manifold

The experiments on the Grassmannian manifolds are designed to tackle the problem of recognition from videos or image-sets. In this context, a video or an image-set is divided into 3D blocks, followed by describing each block by a linear subspace through SVD decomposition.

Face Recognition:
As our first experiment on Grassmannian manifolds, we tackled the task of video-based face recognition. To this end, we considered the YouTube Celebrity dataset [28] which contains 1910 videos of 47 people (see Fig. 2 for examples). The large diversity of poses, illumination, and facial expressions in addition to high compression ratio of face images provide a significant challenges in this dataset.

For our evaluation, we followed the setup used in the method of Adaptive Deep Network Template (ADNT) by Hayat *et al*. [23]. More specifically, from each video, the face regions are extracted using the tracker of Ross *et al*. [39]. Then, without any further refinement, each face region was divided into 4×4 distinct non-overlapping blocks and the histogram of Local Binary Patterns (LBP) [33] was extracted for each patch and concatenated to form the final frame descriptors.

We note that various evaluation protocols were used by researchers on this dataset. Here, we use the five-fold cross validation protocol introduced in [23], which divides the whole dataset equally (with minimum overlap) into five folds with 9 videos per subject in each fold. Three of the videos were randomly selected for training, while the remaining six were used for testing.

We generated linear subspaces of order 6 by grouping features of every 6 consecutive frames. Therefore, each local descriptor belongs to $\mathcal{G}^6_{928}$. The second column of Table 3 summarizes the average recognition rates and the standard deviations of all the studied methods.

We note that the R-VLAD with both geodesic and projection metric comfortably outperforms the state-of-the-art

ADNT algorithm. The maximum accuracy of 79.9% is achieved by R-VLAD using projection metric with a linear SVM classifier.

Dynamic Texture Classification:
We performed an experiment to classify videos of dynamic textures using the Dyntex++ dataset [14]. Dynamic textures are videos of moving scenes (smoke, waves, high way, forest fire, etc.) that exhibit certain stationarity properties in time domain. The DynTex++ dataset contains 3600 ($50 \times 50 \times 50$) videos of moving scenes in 36 classes.

To obtain local Grassmannian descriptors, each video is decomposed into 3D blocks of size $15 \times 15 \times 15$ with spatial (temporal) overlap of 5 pixels (frames). The 3D block was then described by grouping its internal frames and describing each with the 3D extension of the Local Binary Pattern (LBP) [33], namely LBP Three Orthogonal Planes (LBP-TOP) [53]. For each 3D block and from the LBPTOP features, we extracted a subspace of dimension 6 using SVD. This resulted in having local descriptors on $\mathcal{G}^6_{177}$. In total, we extracted 512 subspaces from each video.

For this experiment, we followed the evaluation protocol used in [4]. That is, half of the videos of each class were randomly chosen as training data and the remaining ones were used as test data. The process of random selection was repeated 10 times and the average accuracy along standard deviations are reported in the third column of Table 3.

Table 3 shows that the proposed R-VLAD outperforms the state-of-the-art by more than 5 percentage points. R-VLAD with projection metric equipped with a linear SVM classifier achieves the highest recognition rate of 97.6%. Compared to Log-Euclidean solutions, again R-VLAD is preferable though the gap is not as big as that of the previous experiment.

Dynamic Scene Recognition:
As the last experiment, we considered the task of scene recognition from the videos using UPenn dataset [10] (see Fig. 2 for example classes). The UPenn dataset consists of 420 videos of natural scenes spanning 14 categories. The videos are obtained from various sources including personal footage captures and online repositories such as YouTube. Moreover, significant differences in image resolution and appearance, frame rate, scale, viewpoint, and illumination conditions exist in this dataset.

To extract local Grassmannian points, we followed the setup used in the previous experiment and represented each video using 6 dimensional subspaces extracted from $15 \times 15 \times 15$ subblocks (using LBPTOP features). Leave-one-video-out scenario is the standard evaluation protocol on this dataset. To the best of our knowledge, the recent Bag of Spatiotemporal Energy (BoSE) method of Feichtenhofer *et al*. [13] has achieved the highest accuracy on this dataset.

The recognition accuracies for all the studied methods

Figure 2: Examples of YouTube celebrity and UPENN datasets.

Table 3: Recognition accuracies for the YouTube Celebrities [28], Dyntex++ [14], and UPENN [10] datasets.

Method	YouTube	Dyntex++	UPENN
Previous Best	71.4 ± 5.1 [23]	92.4 [4]	96.2 [13]
BOW$_G$-NN	60.3 ± 5.4	92.0 ± 0.7	81.2
BOW$_G$-SVM	64.5 ± 5.1	92.4 ± 0.5	92.9
BOW$_{LE}$-NN	50.6 ± 3.9	80.6 ± 0.9	75.2
BOW$_{LE}$-SVM	55.3 ± 2.9	81.1 ± 0.5	92.6
VLAD$_{LE}$-NN	62.1 ± 1.6	93.1 ± 0.6	86.4
VLAD$_{LE}$-SVM	65.2 ± 2.8	93.3 ± 0.4	96.9
R-VLAD$_G$-NN	75.5 ± 3.4	96.4 ± 0.4	90.0
R-VLAD$_G$-SVM	75.6 ± 2.5	96.7 ± 0.3	99.5
R-VLAD$_P$-NN	78.5 ± 3.6	96.9 ± 0.4	91.2
R-VLAD$_P$-SVM	$\mathbf{79.9 \pm 3.6}$	$\mathbf{97.6 \pm 0.4}$	**99.8**

are shown in the last column of Table 3. The results are self-explanatory. The R-VLAD equipped with a linear SVM classifier achieves the highest accuracy, outperforming the state-of-the-art by more than 3 percentage points. Notably, the Log-Euclidean VLAD is performing slightly better than the state-of-the-art method of Feichtenhofer *et al*. [13].

7. Main Findings and Future Directions

Inspired by the recent success of compact descriptors in Euclidean spaces and superior discriminative power of the descriptors on Riemannian manifolds, in this paper we proposed R-VLAD, the Riemannian version of the conventional *Vector of Locally Aggregated Descriptors* (VLAD). In addition to the comprehensive formulation, we devised manifold-specific versions of the R-VLAD on the SPD and Grassmannian. An extensive set of successful experiments on several challenging vision tasks including action recognition from MoCaP data, face recognition from videos, and dynamic scene categorization supported our method. Since our formulation allows us to utilize any metric to construct VLAD, we plan to investigate how more robust metrics (*e.g.*, ℓ_1) can be employed in conventional VLAD. We are also interested to extend our framework to other types of Riemannian structures such as Kendall shape manifolds.

Acknowledgements

Authors would like to thank Professor Richard Hartley for fruitful discussions.

This research was supported under Australian Research Councils Discovery Projects funding scheme (project DP150104645).

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2008. 5

[2] R. Arandjelovic and A. Zisserman. All about vlad. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1578–1585. IEEE, 2013. 2

[3] 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. 6

[4] M. Baktashmotlagh, M. Harandi, B. C. Lovell, and M. Salzmann. Discriminative non-linear stationary subspace analysis for video classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(12):2353 – 2366, 2014. 8

[5] E. Begelfor and M. Werman. Affine invariance revisited. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2087–2094, 2006. 5

[6] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 6

[7] D. Chen, S. Tsai, V. Chandrasekhar, G. Takacs, H. Chen, R. Vedantham, R. Grzeszczuk, and B. Girod. Residual enhanced visual vectors for on-device image matching. In *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pages 850–854. IEEE, 2011. 2

[8] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos. Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(9):2161–2174, 2012. 5

[9] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2

[10] K. G. Derpanis, M. Lecce, K. Daniilidis, and R. P. Wildes. Dynamic scene understanding: The role of orientation features in space and time in scene classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1306–1313. IEEE, 2012. 2, 8

[11] M. Faraki, M. T. Harandi, and F. Porikli. Material classification on symmetric positive definite manifolds. In *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 749–756, 2015. 1

[12] M. Faraki, M. T. Harandi, A. Wiliem, and B. C. Lovell. Fisher tensors for classifying human epithelial cells. *Pattern Recognition (PR)*, 47(7):2348–2359, 2014. 1

[13] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Bags of spacetime energies for dynamic scene recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2014. 2, 8

[14] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In *Proc. European Conference on Computer Vision (ECCV)*, pages 223–236, 2010. 2, 8

[15] A. Goh and R. Vidal. Clustering and dimensionality reduction on Riemannian manifolds. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7. IEEE, 2008. 3

[16] G. H. Golub and C. F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. 6

[17] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Proc. European Conference on Computer Vision (ECCV)*, pages 392–407. Springer, 2014. 1, 2

[18] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proc. Int. Conference on Computer Vision (ICCV)*, volume 2, pages 1458–1465. IEEE, 2005. 2

[19] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar. Multiresolution histograms and their use for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(7):831–847, 2004. 2

[20] J. Hamm and D. D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *Proc. Int. Conference on Machine Learning (ICML)*, pages 376–383. ACM, 2008. 2, 5

[21] M. Harandi, M. Salzmann, and F. Porikli. Bregman divergences for infinite dimensional covariance matrices. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014. 7

[22] M. Harandi, C. Sanderson, C. Shen, and B. Lovell. Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 3120–3127. IEEE, 2013. 1, 5

[23] M. Hayat, M. Bennamoun, and S. An. Learning non-linear reconstruction models for image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1915–1922. IEEE, 2014. 2, 7, 8

[24] J. Ho, Y. Xie, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *Proc. Int. Conference on Machine Learning (ICML)*, pages 1480–1488, 2013. 3

[25] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2555–2562. IEEE, 2013. 2

[26] H. Jégou and O. Chum. Negative evidences and cooccurences in image retrieval: The benefit of pca and whitening. In *Proc. European Conference on Computer Vision (ECCV)*, pages 774–787. Springer, 2012. 2

[27] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into com-

pact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(9):1704–1716, 2012. 1, 2

[28] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 2, 7, 8

[29] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178. IEEE, 2006. 2

[30] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *Int. Journal of Computer Vision (IJCV)*, 43(1):29–44, 2001. 2

[31] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 2

[32] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Mocap database hdm05. *Institut für Informatik II, Universität Bonn*, 2007. 2, 7

[33] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(7):971–987, 2002. 7, 8

[34] X. Peng, L. Wang, Y. Qiao, and Q. Peng. Boosting vlad with supervised dictionary learning and high-order statistics. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Proc. European Conference on Computer Vision (ECCV)*, volume 8691 of *Lecture Notes in Computer Science*, pages 660–674. Springer International Publishing, 2014. 2

[35] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *Int. Journal of Computer Vision (IJCV)*, 66(1):41–66, 2006. 3, 4

[36] F. Perronnin, Z. Akata, Z. Harchaoui, and C. Schmid. Towards good practice in large-scale learning for image classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3482–3489. IEEE, 2012. 2

[37] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 2

[38] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. European Conference on Computer Vision (ECCV)*, pages 143–156. Springer, 2010. 4

[39] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *Int. Journal of Computer Vision (IJCV)*, 77(1-3):125–141, 2008. 7

[40] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988. 2

[41] K. Simonyan, A. Vedaldi, , and A. Zisserman. Deep fisher networks for large-scale image classification. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2013. 1

[42] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 1470–1477. IEEE, 2003. 2

[43] S. Sra. A new metric on the manifold of kernel matrices with application to matrix geometric means. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 144–152, 2012. 1, 4

[44] S. Sra and R. Hosseini. Conic geometric optimisation on the manifold of positive definite matrices. *arXiv:1312.1039*, 2014. 5

[45] R. Subbarao and P. Meer. Nonlinear mean shift over Riemannian manifolds. *Int. Journal of Computer Vision (IJCV)*, 84(1):1–20, 2009. 3

[46] D. Tosato, M. Spera, M. Cristani, and V. Murino. Characterizing humans on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1972–1984, 2013. 1, 2, 6, 7

[47] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(11):2273–2286, 2011. 5

[48] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(10):1713–1727, 2008. 1

[49] R. Vemulapalli, J. K. Pillai, and R. Chellappa. Kernel learning for extrinsic classification of manifold features. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1782–1789. IEEE, 2013. 1

[50] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 3551–3558. IEEE, 2013. 1

[51] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2496–2503. IEEE, 2012. 1

[52] Z. Wang and B. C. Vemuri. An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 223–228. IEEE, 2004. 1, 4

[53] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(6):915–928, 2007. 8