Material Classification on Symmetric Positive Definite 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 tackles the problem of categorizing materials and textures by exploiting the second order statistics. To this end, we introduce the Extrinsic Vector of Locally Aggregated Descriptors (E-VLAD), a method to combine local and structured descriptors into a unified vector representation where each local descriptor is a Covariance Descriptor (CovD). In doing so, we make use of an accelerated method of obtaining a visual codebook where each atom is itself a CovD. We will then introduce an efficient way of aggregating local CovDs into a vector representation. Our method could be understood as an extrinsic extension of the highly acclaimed method of Vector of Locally Aggregated Descriptors [17] (or VLAD) to CovDs. We will show that the proposed method is extremely powerful in classifying materials/textures and can outperform complex machineries even with simple classifiers.

***keywords**: Riemannian manifolds, Region covariance descriptor, Vector of locally aggregated descriptors, Material classification, Texture recognition.*

1. Introduction

In this paper we propose a method for generating compact and discriminative representations from localized structured descriptors in the form of Covariance Descriptors (CovDs) for categorizing materials and textures. Recently, material categorization based on images has received growing attention, mainly because of its broad applications [3]. For example, in recycling centers it is required to discriminate various types of household wastes automatically (*e.g.*, paper from plastic or fabric). As another example, an autonomous robot needs to judge where to step on, and its decision is affected by the type of the materials lying on the floor. The dominant trend in classifying materials is to utilize textural information [21, 37]. However, textural cues in materials occur at finer scales as compared to ordinary texture images, which makes the two problems correlated but not identical [21].

CovD [35] and its spatio-temporal extension, *i.e.*, Cov3D [29], have been employed successfully in various vision applications such as pedestrian detection, object tracking, classifying human epithelial cells, and analyzing diffusion tensor images (see for example [8, 25, 35, 14, 16, 7] and references therein). Despite their popularity, most of previous studies was devoted to the scenarios where a holistic representation of images or videos sufficed for addressing the problem in hand. One exception is the method of ARray of COvariance matrices (ARCO) proposed by Tosato *et al*. [34]. In ARCO, a set of local CovDs is extracted from an image or video. To perform classification, each local CovDs is assessed by a separate classifier and majority voting is used to determine the final result. Nevertheless, training multiple classifiers on CovDs is expensive and the resulting system requires retraining if new classes or data are introduced.

In contrast to previous attempts, in this paper we propose an unsupervised method for creating a compact and discriminative vector representation from a set of local CovDs (extracted from an image or video). Our idea here is inspired by the recent and acclaimed method of Vector of Locally Aggregated Vectors (VLAD) [17] and extends the original VLAD method to work with CovDs. This is ofcourse not a trivial extension as CovDs are Symmetric Positive Definite (SPD) matrices with a well-known Riemannian structure. As shown by a large body of recent studies [8, 13, 12, 35, 14, 16], machineries that exploit the Riemannian geometry of CovDs should be preferred to the the ones that make use of Euclidean geometry in handling CovDs.

To this end, we first elaborate on an accelerated method of learning a codebook on SPD, or tensor manifolds[1]. We then introduce our main contribution, the Extrinsic version of VLAD or E-VLAD. Compared to a BoW model on tensor manifolds, E-VLAD encodes additional information about the distribution of local descriptors in a compact rep-

[1]The word tensor manifold in this work is referred to the manifold of symmetric positive definite matrices and should not be confused with the rigorous mathematical definition of tensors.

resentation with relatively the same computational burden.

Our experiments show the superiority of the proposed method against several state-of-the-art methods such as the texton approach of Varma and Zisserman (VZ) [36, 37], the BoW approach of Sharan *et al*. [30], the augmented Latent Dirichlet Allocation (aLDA) method of Liu *et al*. [21], and the Local Higher-order Statistics (LHS) method of Sharma *et al*. [32] on various challenging datasets such as FMD [31] and KTH-TIPS2-a [3] material, and Dyntex++ [9] dynamic texture databases.

2. Related Work

Materials (and textures) can be analyzed based on the reflectance properties of surfaces. While this school of thought has received considerable attention in computer graphics, successful methods rely on extra (and to some extent restrictive) constraints on surfaces, illuminations, and structure of materials [27].

Different from methods that rely on reflectance properties, several studies propose filter banks for analyzing texture and material images. Two notable examples are Leung and Malik (LM) and Maximum Response filter banks [36, 19]. Nevertheless, a more recent study by Varma and Zisserman demonstrated that materials can be also classified using the statistical properties of pixels [37]. In particular, it is shown that joint distribution of intensity values over compact neighborhoods can outperform filter banks with large supports.

The work by Varma and Zisserman inspired several studies to explore the concept of BoW for material classification. The underlying idea is to utilize small images with rich statistical properties to encode texture or material images. One example is the method proposed by Liu *et al*. [22] for rotationally invariant material/texture classification. In [22], a set of random measurements from sorted pixel differences is used in a BoW model for classification. In [21], low and mid-level features are used to successfully capture appearance of materials. Several codebooks are then formed from extracted features and then combined using an extended Latent Dirichlet Allocation (aLDA) model.

In this paper, we propose to create BoW models using CovDs which encode the second order statistics of textural information. We will show that BoW models on CovDs are astonishingly rich and can outperform complex machineries even with simple classifiers. Nevertheless, since CovDs are lying on a Riemannian manifold, generating BoW models is not trivial.

3. Background

In this section, we review basics of Riemannian geometry, manifold of real SPD matrices, and associated metrics. Throughout this paper, $\mathcal{S}_{++}^d$ denotes the space of $d \times d$ SPD matrices.

3.1. Riemannian Geometry

A *manifold* $\mathcal{M}$ is a Hausdorff topological space which locally resembles the Euclidean space. More specifically, for each point and a neighborhood around it, there exists a homeomorphic (one-to-one, onto, and bidirectional continuous map) to an open set in $\mathbb{R}^m$ for some m. The *tangent space* attached to a point $\boldsymbol{P}$ on a differentiable manifold, $T_{\boldsymbol{P}}\mathcal{M}$, is a vector space that consists of the tangent vectors of all possible curves passing through $\boldsymbol{P}$.

A *Riemannian manifold* is a differential manifold with a metric defined on its tangent spaces. The metric enables us to define lengths and angles and is usually chosen such that some sort of robustness to geometrical transformations is achieved. Points on a Riemannian manifold are connected through smooth curves. The curve with the minimum length is called the *geodesic* curve and its length is the geodesic distance between points.

Two operators, namely the *exponential map* and the *logarithm map* are defined to switch between the manifold and its tangent space at $\boldsymbol{P}$. The exponential operator $\exp_{\boldsymbol{P}}(\cdot) : T_{\boldsymbol{P}}\mathcal{M} \rightarrow \mathcal{M}$, maps a tangent vector Δ to a point $\boldsymbol{X}$ on the manifold. The logarithm map $\log_{\boldsymbol{P}}(\cdot) = \exp_{\boldsymbol{P}}^{-1}(\cdot) : \mathcal{M} \rightarrow T_{\boldsymbol{P}}\mathcal{M}$, is the inverse of the exponential map and maps a point on the manifold to the tangent space at $\boldsymbol{P}$. We note that, the exponential and logarithm maps vary as the point $\boldsymbol{P}$ moves along the manifold.

Geometry of SPD Matrices

The *Affine Invariant Riemannian Metric (AIRM) [25]* is the most popular choice to handle the non-Euclidean structure of SPD matrices and is shown to be advantageous for several applications [35]. For $\boldsymbol{P} \in \mathcal{S}_{++}^d$ and two tangent vectors $\Delta_1, \Delta_2 \in T_{\boldsymbol{P}}\mathcal{M}$, the AIRM is defined as:

$$\begin{aligned} \langle \Delta_1, \Delta_2 \rangle_{\boldsymbol{P}} &\triangleq \langle \boldsymbol{P}^{-1/2}\Delta_1\boldsymbol{P}^{-1/2}, \boldsymbol{P}^{-1/2}\Delta_2\boldsymbol{P}^{-1/2} \rangle \\ &= \mathrm{Tr}\left(\boldsymbol{P}^{-1}\Delta_1\boldsymbol{P}^{-1}\Delta_2\right) \ . \end{aligned} \tag{1}$$

For two $\boldsymbol{X}, \boldsymbol{Y} \in \mathcal{S}_{++}^d$, the geodesic distance induced by AIRM is:

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

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

Despite its appealing features, computing AIRM for two SPD matrices demands eigenvalue decomposition. This incurs hefty computational load for handling a large set of CovDs. In this paper, we use the Stein metric [33], which is a member of the symmetrized Bregman matrix divergences. The asymmetric Bregman divergence for two SPD matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ is defines as

$$\delta_\psi(\boldsymbol{X},\boldsymbol{Y}) \triangleq \psi(\boldsymbol{X}) - \psi(\boldsymbol{Y}) - \langle \nabla_\psi(\boldsymbol{Y}),\ \boldsymbol{X}-\boldsymbol{Y}\rangle\ , \quad (3)$$

where $\langle \boldsymbol{X},\boldsymbol{Y}\rangle = \mathrm{Tr}\left(\boldsymbol{X}^T\boldsymbol{Y}\right)$ and $\psi : \mathcal{S}_{++}^d \to \mathbb{R}$ is a real-valued, strictly convex and differentiable function. While the Bregman divergences exhibit a number of useful properties, their asymmetric behavior is often counter-intuitive and undesirable in practical applications. Therefore, a symmetric version of Bregman divergence is usually preferred.

$$\delta_{JS}(\boldsymbol{X},\boldsymbol{Y}) \triangleq \frac{1}{2}\delta_\psi\left(\boldsymbol{X}, \frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right) + \frac{1}{2}\delta_\psi\left(\boldsymbol{Y}, \frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right)\ . \quad (4)$$

If ψ in (4) is chosen as $-\ln\det(\boldsymbol{X})$ with $\det(\cdot)$ denoting the determinant, we arrive at the symmetric Stein divergence:

$$S(\boldsymbol{X},\boldsymbol{Y}) \triangleq \ln\det\left(\frac{\boldsymbol{X}+\boldsymbol{Y}}{2}\right) - \frac{1}{2}\ln\det\Big(\boldsymbol{X}\boldsymbol{Y}\Big), \quad (5)$$

Interestingly $\delta_S(\cdot,\cdot) \triangleq \sqrt{S(\cdot,\cdot)}$ is a valid metric and though computationally less demanding, is related to AIRM in many aspects. Similar to AIRM, $\delta_S^2(\cdot,\cdot)$ is affine invariant, endows a sandwiching property (*i.e.*, $\alpha_1\delta_S^2(\boldsymbol{X},\boldsymbol{Y}) \leq \delta_R^2(\boldsymbol{X},\boldsymbol{Y}) \leq \alpha_2\delta_S^2(\boldsymbol{X},\boldsymbol{Y})$) and behaves very similarly to AIRM along geodesics on the manifold. More specifically, This provides our motivation for addressing problems on $\mathcal{S}_{++}^d$ via the Stein metric.

4. Learning Riemannian codebooks

Learning a codebook is crucial to our proposed method (*i.e.* E-VLAD). In this section, we elaborate on possible learning methods to generate a codebook specific to SPD manifolds. Formally, given a set of training samples $\{\boldsymbol{X}_i\}_{i=1}^N, \boldsymbol{X}_i \in \mathcal{S}_{++}^d$, we seek to estimate k clusters $C_1, C_2, \cdots, C_k$ with centers $\{\boldsymbol{S}_i\}_{i=1}^k$ such that the sum of squared distances over all clusters is minimized, *i.e.*:

$$\underset{C_1,C_2,\cdots,C_k}{min} \sum_{i=1}^{k} \sum_{\boldsymbol{X}_j \in C_i} \delta^2(\boldsymbol{X}_j, \boldsymbol{S}_i), \quad (6)$$

where δ is a metric on $\mathcal{S}_{++}^d$.

In the most straightforward case, one can neglect the geometry of SPD matrices and vectorize training data, *i.e.*, $\boldsymbol{X}_i$ to learn a codebook. As a result, codebook learning becomes a trivial task and can be achieved by applying for example K-means algorithm on vectorized data. More specifically, the resulting clusters are determined by computing the arithmetic mean of nearest training vectors to that cluster.

Despite its simplicity, several studies argue against exploiting Euclidean geometry and vector form of SPD matrices for inference [25, 35]. For example, as shown by Pennec *et al.* [25] the determinant of the weighted mean could become greater than samples' determinants, an undesirable outcome known as swelling effect [1]. Therefore, geometry of SPD matrices should be considered in creating the codebook.

To benefit from the Riemannian geometry, it is possible to replace the arithmetic mean with Karcher (Fráchet) mean [25]. Karcher mean is the point that minimizes the following metric dispersion:

$$\boldsymbol{X}^* = \arg\min_{\boldsymbol{X}} \sum_{i=1}^{m} \delta_R^2(\boldsymbol{X}_i, \boldsymbol{X})\ , \quad (7)$$

where $\delta_R : \mathcal{M}\times\mathcal{M} \to \mathbb{R}^+$ is the associated geodesic distance function.

The discussion of the existence and uniqueness value of the Karcher mean are given in [25]. Since, at the optimum point the gradient is zero, a gradient descent algorithm can be utilized to obtain the mean value. The details of computing the Karcher mean over SPD manifolds are given in [25].

Nevertheless, computing Karcher mean requires switching back and forth between manifold and its tangent spaces which is computationally demanding, especially in our application where a large number of training points is available. Hence, we opt for a faster way of computing a codebook by making use of the Stein metric, *i.e.*, Eqn. (5). As mentioned in § 3.1, the behavior of the Stein metric is very similar to that of AIRM. The main step of learning a codebook is the computation of the centroid for the ith cluster, *i.e.*, $\min_{\boldsymbol{S}_i} \sum_{\boldsymbol{X}_j \in C_i} \delta_S^2(\boldsymbol{X}_j, \boldsymbol{S}_i)$ which can be written as:

$$\underset{\boldsymbol{S}_i}{min} \sum_{\boldsymbol{X}_j \in C_i} \ln\det\left(\frac{\boldsymbol{X}_\mathrm{j}+\boldsymbol{S}_\mathrm{i}}{2}\right) - \frac{1}{2}\ln\det\Big(\boldsymbol{X}_\mathrm{j}\boldsymbol{S}_\mathrm{i}\Big). \quad (8)$$

An iterative, still less demanding way of solving (8) is given by:

$$\boldsymbol{S}_i^{t+1} = \left[\frac{1}{|C_i|}\sum_{\boldsymbol{X}_j \in C_i}\left(\frac{\boldsymbol{X}_j+\boldsymbol{S}_i^t}{2}\right)^{-1}\right]^{-1}. \quad (9)$$

We refer the interested reader to the proofs in [4] for the details. The details of learning an accelerated codebook using the Stein metric is provided in Algorithm 1.

To give the reader some idea about the importance of the accelerated dictionary learning, note that in our experiment on the FMD dataset (see § 6), with near 20,000 training samples of size 135×135, performing 10 iterations of kmeans with AIRM requires about 59 hours on a 3.4GHz CPU, while the accelerated method can create the dictionary in less than 2 hours.

Algorithm 1 Accelerated k-means algorithm over $\mathcal{S}_{++}^d$ for learning the codebook

Input:

- training set $\mathbb{X}=\{\boldsymbol{X}_j\}_{j=1}^N$ from the underlying $\mathcal{S}_{++}^d$ manifold,
- $nIter$, the number of iterations

Output:

- codebook $\mathbb{S}=\{\boldsymbol{S}_i\}_{i=1}^k, \boldsymbol{S}_i \in \mathcal{S}_{++}^d$

1: Initialize the codebook $\mathbb{S}=\{\boldsymbol{S}_i\}_{i=1}^k$ by selecting k samples from $\mathbb{X}$ randomly
2: **for** $t=1 \rightarrow nIter$ **do**
3: Assign each point $\boldsymbol{X}_j$ to its nearest cluster in $\mathbb{S}$ by computing $\ln\det\left(\frac{\boldsymbol{X}_j+\boldsymbol{S}_i}{2}\right)-\frac{1}{2}\ln\det\left(\boldsymbol{X}_j\boldsymbol{S}_i\right),\ 1\leq j\leq N, 1\leq i\leq k$
4: Recompute cluster centers $\{\boldsymbol{S}_i\}_{i=1}^k$ by $\boldsymbol{S}_i^{t+1}=\left[\frac{1}{|C_i|}\sum_{\boldsymbol{X}_j\in C_i}\left(\frac{\boldsymbol{X}_j+\boldsymbol{S}_i^t}{2}\right)^{-1}\right]^{-1}$
5: **end for**

5. Extrinsic Vector of Locally Aggregated Descriptors (E-VLAD) on SPD Manifolds

In the previous section, we elaborated on how an accelerated codebook can be obtained on $\mathcal{S}_{++}^d$. In this section, we provide a detailed description on our proposed encoding method for a set of local descriptors. In other words, having a codebook, $\mathbb{S}=\{\boldsymbol{S}_i\}_{i=1}^k$, at our disposal , we seek to group a set of CovDs (or equivalently Cov3Ds), $\mathbb{Q}=\{\boldsymbol{Q}_i\}_{i=1}^p$, extracted from a query image (video) in order to find a rich representation.

Jégou *et al*. [17] proposed the Vector of Locally Aggregated Descriptors (VLAD) for the task of large-scale image search. In VLAD, a codebook is learned by K-means on training samples, and subsequently the extracted local descriptors of an image are assigned to the closest codeword. Then, differences of each codeword to its descriptors are accumulated. Finally, the accumulated vectors of each codeword are concatenated and normalized.

Since, SPD matrices do not form a closed set under normal matrix subtraction, *i.e*., subtracting two SPD matrices does not result in another SPD matrix, the VLAD framework can not be readily extended to the space of tensors. Our idea here is to simply embed the manifold into a vector space. To this end, we make use of a mapping from $\mathcal{S}_{++}^d$ into the space of symmetric matrices by the principal matrix logarithm. We are motivated by the fact that, there always exists a unique, real and symmetric logarithm for any SPD matrix, which can be obtained by principal matrix logarithm. Moreover, $\log(\cdot)$ on $\mathcal{S}_{++}^d$ is diffeomorphism (a one-to-one, continuous, differentiable mapping with a continuous, differentiable inverse). Formally,

Theorem 1: $\log(\cdot) : \mathcal{S}_{++}^d \rightarrow Sym(d)$ is C^∞ and therefore both $\log(\cdot)$ and its inverse $\exp(\cdot)$ are smooth, *i.e*., they are diffeomorphisms.

Algorithm 2 The proposed E-VLAD algorithm

Input:

- $\mathbb{Q}=\{\boldsymbol{Q}_t\}_{t=1}^p$, CovDs extracted from a query image,
- codebook $\mathbb{S}=\{\boldsymbol{S}_i\}_{i=1}^k, \boldsymbol{S}_i \in \mathcal{S}_{++}^d$

Output:

- $\boldsymbol{EV}(\mathbb{Q})$ the E-VLAD representation of $\mathbb{Q}$

1: Compute log-Euclidean representation of $\mathbb{S}$ using $\boldsymbol{s}_i=\mathrm{Vec}(\log(\boldsymbol{S}_i))$
2: Compute log-Euclidean representation of $\mathbb{Q}$ using $\boldsymbol{q}_t=\mathrm{Vec}(\log(\boldsymbol{Q}_t))$
3: **for** $i=1 \rightarrow k$ **do**
4: Find C_i, all nearest CovDs from $\mathbb{Q}$ to $\boldsymbol{S}_i$ using Eqn. (5)
5: Compute i-th accumulator, $\boldsymbol{v}_i=\sum_{\boldsymbol{q}_j\in C_i}\boldsymbol{q}_j-\boldsymbol{s}_i$
6: **end for**
7: Concatenate the resulting accumulators to form the final descriptor, *i.e*., $\boldsymbol{EV}(\mathbb{Q})=\left[\boldsymbol{v}_1^T,\boldsymbol{v}_2^T,\cdots,\boldsymbol{v}_k^T\right]^T$

Proof: We refer the reader to [1] for the proof of this theorem.

Embedding into the space of symmetric matrices through principal logarithm can also be understood as embedding $\mathcal{S}_{++}^d$ into its tangent space at identity matrix. Since tangent spaces form a vector space, then we are able to employ Euclidean tools to tackle the problem in hand. We note that our approach here is an extrinsic approach, *i.e*., it depends on the embedding Euclidean space.

Given an SPD matrix $\boldsymbol{A}$, its log-Euclidean vector representation, $\boldsymbol{a}$, is unique and defined as $\boldsymbol{a}=\mathrm{Vec}\left(\log(\boldsymbol{A})\right)$ where the $\mathrm{Vec}\left(\boldsymbol{B}\right),\ \boldsymbol{B}\in Sym(d)$ is:

$$\mathrm{Vec}\left(\boldsymbol{B}\right)=\left[b_{1,1},\sqrt{2}b_{1,2},\cdots,\sqrt{2}b_{1,d},b_{2,2},\sqrt{2}b_{2,3},\cdots,b_{d,d}\right]^T. \tag{10}$$

Let $\mathbb{Q}=\{\boldsymbol{Q}_i\}_{i=1}^p$ and $\mathbb{S}=\{\boldsymbol{S}_i\}_{i=1}^k$, be a set of CovDs (extracted from a query image) and codewords (obtained by intrinsic K-means 1), respectively. For each codeword $\boldsymbol{S}_i$, the resulting accumulated differences is given by:

$$\boldsymbol{v}_i=\sum_{\boldsymbol{Q}_j\in C_i}\mathrm{Vec}(\log(\boldsymbol{Q}_j))-\mathrm{Vec}(\log(\boldsymbol{S}_i)) \tag{11}$$

where CovDs belonging to a codeword $\boldsymbol{S}_i$ (*i.e*., C_i) can be found by the Stein metric. Final descriptor $\boldsymbol{EV}$, is obtained by concatenating k $\boldsymbol{v}_i$ vectors associated with codewords. That is, $\boldsymbol{EV}(\mathbb{Q})=\left[\boldsymbol{v}_1^T,\boldsymbol{v}_2^T,\cdots,\boldsymbol{v}_k^T\right]^T$. Algorithm 2 assembles all the above details into pseudo-code for E-VLAD.

6. Experiments

We start this section by introducing databases used in our experiments. We then evaluate the performance of the pro-

posed E-VLAD against several state-of-the-art approaches on material categorization and dynamic texture classification.

In our experiments a set of overlapping blocks/cubes is extracted from images/videos and the CovD/Cov3D for each block/cube is obtained. Each E-VLAD descriptor is normalized in two steps. First, a *power normalization* is performed on each element of the E-VLAD based on the recommendation of Jegou *et al*. [17]. This is to avoid having concentrated distribution around zero. The transfer function for *power normalization* is: $y : \mathbb{R} \rightarrow \mathbb{R},\ y(x) = sign(x)\sqrt{|x|}$, where x is an element of E-VLAD and $|\cdot|$ denotes absolute value. The power normalization is followed by ℓ_2 normalization. To classify signatures, Support Vector Machines (SVM) [2] are employed.

In all the reported experiments, the size of the dictionary for E-VLAD is set to 16. The aforementioned value is obtained empirically as a trade-off between the complexity of encoding and performance of algorithms. We will later provide a diagram discussing the performance of Riemannian version of Bag of Words (R-BoW) against baseline methods for various dictionary sizes. Similar to E-VLAD, R-Bow utilizes Algorithm 1 to generate its codebook. As for the image representation, we followed the simplest form of BoW model and assigned CovDs to their closest codewords. The comparisons were done using the Stein metric, followed by ℓ_2 normalization in the end.

We also provide the results of Log-Euclidean VLAD ($VLAD_{LE}$), which can be readily considered as an extension of VLAD into the space of SPD matrices. We note that VLAD_{LE} differs from E-VLAD in the sense that in VLAD_{LE} CovDs/Cov3Ds are first mapped to the tangent space at the identity matrix via Eqn. 10. In contrast, in E-VLAD, the logarithm mapping is used after proper assignment of input CovDs/Cov3Ds to the codebook. We will show experimentally that E-VLAD consistently outperforms VLAD_{LE} which we conjecture is due to better exploitation of Riemannian geometry.

6.1. Databases

In this subsection, we introduce the databases used in our experiments. Sample images/frames are shown in Fig. 1.

Flicker Material Database: Flicker Material Database [31] (FMD) collected from Flicker photos of daily life material categories. It is designed from real world examples with large scale and intra-class variations. FMD images are presented in 10 material categories and each category has 100 images.

UIUC Material Database: UIUC database [20] contains 18 classes of complex material categories in local scale "taken in the wild". The images are mainly selected to have more (compared to FMD) geometric fine-scale details.

Figure 1: Sample images from databases used in our experiments. From top to bottom images selected from FMD [31], UIUC [20], KTH-TIPS [15], KTH-TIPS2-a [3], and DynTex++ [9] databases.

KTH-TIPS Material Database: This database is specifically designed for material classification task [15]. It contains 810 images from 10 different classes captured at 9 scales, 3 different illumination directions, and 3 different poses. Images in the database have no obvious rotation.

KTH-TIPS2-a Material Database: KTH-TIPS2-a database [3] provides a considerable extension to KTH-TIPS database. It consists of 4397 images in 11 classes with four samples per class. The images are photographed at 9 scales, 4 different illumination conditions, and 3 poses.

DynTex++ Dynamic Texture Database: DynTex++ database [9] contains videos of moving scenes in 36 classes. Each class is comprised of 100 ($50 \times 50 \times 50$) pre-processed videos.

Except for KTH-TIPS2-a, we split the databases into even size training and test sets. This was done by randomly assigning half of the instances of each class to the training data and using the remaining images as the test set. The random split was repeated 14 times for FMD (as proposed by Sharan *et al*. [30]) and 10 times for the others. For evaluation on KTH-TIPS2-a, we followed the standard protocol used in [3, 32] where three samples of each class are considered for training and the remaining sample is used for testing.

6.2. Application to Material Categorization

We first elaborate how CovDs are extracted from images for material categorization. A feature vector is assigned to sampled pixels in the image through using dense SIFT [38] features. More specifically, image pixels are first sampled

Table 1: Comparisons between the proposed approaches to the state-of-the-art methods on Material and dynamic texture classification databases. Correct Classification Rate (in %) is reported here.

Database	VZ[36]	aLDA[21]	Harandi[11]	Sharan[30]	Liao[20]	VZ[37]	Liu[22]	Caputo[3]	LHS[32]	PD-LBP[28]	$VLAD_{LE}$	**E-VLAD**
FMD	23.8	44.6	51.4	55.6	29.9	-	-	-	-	-	55.4	**56.5**
UIUC	-	-	-	-	43.5	-	-	-	-	-	43.4	**60.1**
KTH-TIPS	-	-	-	-	-	92.4	99.1	-	-	-	99.1	**99.5**
KTH-TIPS2-a	-	-	-	-	-	-	-	71.0	73.0	-	71.4	**76.1**
DynTex++	-	-	-	-	-	-	-	-	-	92.4	89.2	**92.9**

at a dense grid with 3 pixel spacing and then described by

$$F_{(x,y)} = \left[I_R(x,y), I_G(x,y), I_B(x,y), \left|\frac{\partial I}{\partial x}\right|, \left|\frac{\partial I}{\partial y}\right| , \left|\frac{\partial^2 I}{\partial x^2}\right|, \left|\frac{\partial^2 I}{\partial y^2}\right|, |H_1(x,y)|, \cdots, |H_{128}(x,y)| \right], \quad (12)$$

where $I_c(x,y),\ c \in \{R,G,B\}$ denotes color information at position (x,y), $I(x,y)$ is the gray-value intensity, $\left|\frac{\partial I}{\partial x}\right|$ and $\left|\frac{\partial I}{\partial y}\right|$ are magnitude of gradients along x and y directions, $\left|\frac{\partial^2 I}{\partial x^2}\right|$ and $\left|\frac{\partial I^2}{\partial y^2}\right|$ are magnitude of Laplacians along x and y directions, and $H_1(x,y)$ to $H_{128}(x,y)$ are bin values of the standard SIFT descriptor computed at x, y.

Therefore, Each region is described by a 135×135 covariance matrix formed from the aforementioned features. We compare E-VLAD against the state-of-the-art methods proposed in [21, 30, 20, 3, 32, 22]. Beside the state-of-the-art methods, we use Varma-Zisserman's (VZ) algorithm [37, 36] as a baseline here. Loosely speaking, the VZ method clusters 5×5 pixel gray-scale patches as codewords, obtains histogram of the codewords, and performs classification by nearest neighbor classifier.

In [21], authors considered the FMD database and proposed complex low and mid-level features in a Bayesian framework for material classification. To this end, features such as color, Gabor [18], SIFT [23], micro-Gabor, micro-SIFT, curvature, HOG [6] along the *normal* and *tangent* directions of edges were extracted and combined using augmented Latent Dirichlet Allocation (aLDA) [21]. In a discriminative approach, Sharan *et al*. [30] use the same set of features in a BoW framework and SVM as classifier. Harandi *et al*. [11] perform classification using sparse coding on $\mathcal{S}^d_{++}$ by embedding the space of SPD matrices into Hilbert spaces.

Liao *et al*. [20], propose a more general approach for categorizing materials using both FMD and UIUC databases. In their method, geometric details are extracted from intrinsic material components by a non-parametric patch-based filter. Since UIUC is a very recent database, very few results are available for it.

On KTH-TIPS database, Liu *et al*. [22] extract a set of random measurements from sorted pixel differences and embed them into a BoW model. In [5], texture histograms are obtained from a visual vocabulary of Basic Image Features [10] computed at every pixels at four scales.

On KTH-TIPS2-a database [3], Caputo *et al*. [3] proposed a 3-scale LBP [24] descriptor, and Sharma *et al*. [32] proposed to make use of Local Higher order Statistics (LHS) on image patches for classification.

In Table 1, we compare the performance of the aforementioned methods against E-VLAD. On all databases, E-VLAD approach obtains the highest accuracy. We note that, unlike the methods proposed in [21] and [30] for classifying FMD images, our E-VLAD is not biased to a specific database. On UIUC, the difference between E-VLAD and the method proposed by Liao *et al*. [20], exceeds 16 percentage points. On KTH-TIPS, E-VLAD achieves the highest accuracy of 99.5% and outperforms the other methods. On KTH-TIPS2-a, E-VLAD is more than 3 percentage points better than the closest competitor, *i.e*., the LHS method. More importantly, E-VLAD outperforms $\mathrm{VLAD_{LE}}$ by a large support in all reported results. As mentioned earlier, we conjecture that this stems from better exploitation of Riemannian geometry in obtaining the codebook and signature for E-VLAD.

To show the effectiveness of the proposed descriptor, we performed a further experiment on FMD database. To this end, we compared R-Bow against baseline methods, namely SIFT [23] and LBP [24] in a BoW framework for various codebook sizes in Fig. 2. R-BoW outperforms the baseline methods for all the codebooks.

6.3. Application to Dynamic Texture Classification

In this part, we assess and contrast the proposed method for the task of dynamic texture classification which is closely related to material classification. To this end, we performed an extra experiment on DynTex++ [9] dynamic texture database.

To generate points on the manifold, we chose spatio-temporal Gabor filter banks with moving Gaussian envelope proposed previously for motion analysis [26]. The response of a spatio-temporal Gabor filter centered at (x,y,t) with speed (pixels per frame) υ, orientation θ, and phase offset φ

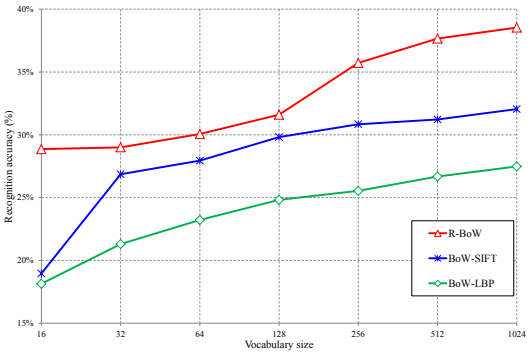

Figure 2: Performance versus the size of dictionary for BoW-LBP, BoW-SIFT, and R-BoW.

is defined as:

$$g_{(\upsilon,\theta,\varphi)}(x,y,t) = \frac{\gamma}{2\pi\sigma^2} exp(\frac{-((\overline{x}+\upsilon_c t)^2+\gamma^2\overline{y}^2)}{2\sigma^2}) \cdot cos(\frac{2\pi}{\lambda}(\overline{x}+\upsilon t)+\varphi) \cdot \frac{1}{\sqrt{2\pi\tau}} exp(\frac{-(t-\mu_t)^2}{2\tau^2}) \ , \quad (13)$$

where $\overline{x} = xcos(\theta)+ysin(\theta)$ and $\overline{y} = -xsin(\theta)+ycos(\theta)$. The parameters of this filter (*i.e.*, γ, λ, φ, τ, and μ) were chosen based on the recommendations of [26]. Each pixel at location (x,y,t) in a spatio-temporal region is described by:

$$f_{(x,y,t)} = \Big[I(x,y,t), \left|\frac{\partial I}{\partial x}\right|, \left|\frac{\partial I}{\partial y}\right|, \left|\frac{\partial I}{\partial t}\right|, |g_{(0,0,\varphi)}(x,y,t)| , |g_{(0,1,\varphi)}(x,y,t)|, ..., |g_{(0,\theta,\varphi)}(x,y,t)| , |g_{(\upsilon,0,\varphi)}(x,y,t)|, ..., |g_{(\upsilon,\theta,\varphi)}(x,y,t)| \Big] \ , \quad (14)$$

We used a combination of 4 speeds and 4 orientations to extract covariance matrices. Therefore, each spatio-temporal region is described by a 20×20 Cov3D. In Table 1 the proposed method is compared against the recent Patch Dependent learning-based LBP (PD-LBP) [28] method. Table 1 indicates that encoding Cov3Ds using E-VLAD outperforms PD-LBP. We also note that E-VLAD is significantly better than $\mathrm{VLAD}_{\mathrm{LE}}$.

7. Conclusions and Future Work

Tackling the task of material classification, in this paper we have introduced an approach to extend Vector of Locally Aggregated Descriptors (VLAD) [17] to the space of Symmetric Positive Definite (SPD) matrices or tensors. In doing so, we followed the concept used in several state-of-the-art methods in material/texture classification (*e.g.*, [37]), which suggests that rich descriptors for both tasks should encode distribution of intensity values over compact neighborhoods. Since Covariance Descriptors (CovD) [35] encode second order statistics, it is natural to exploit them for material classification. The difficulty here comes from the fact that CovDs are SPD matrices and naturally form a Riemannian manifold. The Riemannian structure obviously hindrances employing methods developed in Euclidean spaces to work successfully with CovDs.

To this end, we made use of a special type of the Bregman matrix divergence and introduced an accelerated version of intrinsic k-means algorithm. We then proposed to embed the SPD manifolds into a Euclidean space via a diffeomorphism to extend VLAD to its Riemannian version, *i.e.*, E-VLAD. Our tests showed that E-VLAD consistently outperforms state-of-the-art methods even with simple linear classifiers. Our next goal is to study the performance of the proposed methods on other recognition tasks. Moreover, we are exploring how an intrinsic version of VLAD algorithm can be devised on SPD manifolds.

Acknowledgements

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.

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.

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

[3] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *Proc. Int. Conference on Computer Vision*, volume 2, pages 1597–1604, 2005.

[4] 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*, 35(9):2161–2174, 2012.

[5] M. Crosier and L. D. Griffin. Using basic image features for texture classification. *Int. Journal of Computer Vision*, 88(3):447–460, 2010.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[7] M. Faraki and M. Harandi. Bag of riemannian words for virus classification. *Case Studies in Intelligent Computing: Achievements and Trends*, pages 271–284, 2014.

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

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

[10] L. D. Griffin and M. Lillholm. Feature category systems for 2nd order local image structure induced by natural image statistics and otherwise. In *Electronic Imaging 2007*, pages 649209–649209. International Society for Optics and Photonics, 2007.

[11] M. Harandi, R. Hartley, B. Lovell, and C. Sanderson. Sparse coding on symmetric positive definite manifolds using bregman divergences. *arXiv preprint arXiv:1409.0083*, 2014.

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

[13] M. T. Harandi, M. Salzmann, and R. Hartley. From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In *Proc. European Conference on Computer Vision*, pages 17–32. Springer International Publishing, 2014.

[14] M. T. Harandi, C. Sanderson, A. Wiliem, and B. C. Lovell. Kernel analysis over riemannian manifolds for visual recognition of actions, pedestrians and textures. In *IEEE Workshop on the Applications of Computer Vision*, pages 433–439, 2012.

[15] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh. On the significance of real-world conditions for material classification. In *Proc. European Conference on Computer Vision*, pages 253–266. Springer, 2004.

[16] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 2013.

[17] H. Jégou, F. Perronnin, M. Douze, C. Schmid, et al. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

[18] T. S. Lee. Image representation using 2d Gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):959–971, 1996.

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

[20] Z. Liao, J. Rock, Y. Wang, and D. Forsyth. Non-parametric filtering for geometric detail extraction and material representation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 963–970, 2013.

[21] C. Liu, L. Sharan, E. H. Adelson, and R. Rosenholtz. Exploring features in a bayesian framework for material recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 239–246, 2010.

[22] L. Liu, P. Fieguth, D. Clausi, and G. Kuang. Sorted random projections for robust rotation-invariant texture classification. *Pattern Recognition*, 45(6):2405–2418, 2012.

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

[24] 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*, 24(7):971–987, 2002.

[25] X. Pennec. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006.

[26] N. Petkov and E. Subramanian. Motion detection, noise reduction, texture suppression, and contour enhancement by spatiotemporal gabor filters with surround inhibition. *Biological Cybernetics*, 97(5-6):423–439, 2007.

[27] S. C. Pont and J. J. Koenderink. Bidirectional texture contrast function. *Int. Journal of Computer Vision*, 62(1-2):17–34, 2005.

[28] J. Ren, X. Jiang, and J. Yuan. Dynamic texture recognition using enhanced lbp features. In *IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pages 2400–2404, 2013.

[29] A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell. Spatio-temporal covariance descriptors for action and gesture recognition. In *IEEE Workshop on the Applications of Computer Vision*, pages 103–110, 2013.

[30] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson. Recognizing materials using perceptually inspired features. *Int. Journal of Computer Vision*, 103(3):348–371, 2013.

[31] L. Sharan, R. Rosenholtz, and E. Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 9(8):784–784, 2009.

[32] G. Sharma, S. ul Hussain, and F. Jurie. Local higher-order statistics (lhs) for texture categorization and facial analysis. In *Proc. European Conference on Computer Vision*, pages 1–12, 2012.

[33] 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*, pages 144–152, 2012.

[34] D. Tosato, M. Farenzena, M. Spera, V. Murino, and M. Cristani. Multi-class classification on riemannian manifolds for video surveillance. In *Proc. European Conference on Computer Vision*, pages 378–391. Springer, 2010.

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

[36] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *Int. Journal of Computer Vision*, 62(1-2):61–81, 2005.

[37] M. Varma and A. Zisserman. A statistical approach to material classification using image patch exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2032–2047, 2009.

[38] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.