Comparative Evaluation of Action Recognition Methods via Riemannian Manifolds, Fisher Vectors and GMMs: Ideal and Challenging Conditions

Johanna Carvajal, Arnold Wiliem, Chris McCool, Brian Lovell, Conrad Sanderson

University of Queensland, Brisbane, Australia
Queensland University of Technology, Brisbane, Australia
NICTA, Australia
Data61, CSIRO, Australia

Abstract. We present a comparative evaluation of various techniques for action recognition while keeping as many variables as possible controlled. We employ two categories of Riemannian manifolds: symmetric positive definite matrices and linear subspaces. For both categories we use their corresponding nearest neighbour classifiers, kernels, and recent kernelised sparse representations. We compare against traditional action recognition techniques based on Gaussian mixture models and Fisher vectors (FVs). We evaluate these action recognition techniques under ideal conditions, as well as their sensitivity in more challenging conditions (variations in scale and translation). Despite recent advancements for handling manifolds, manifold based techniques obtain the lowest performance and their kernel representations are more unstable in the presence of challenging conditions. The FV approach obtains the highest accuracy under ideal conditions. Moreover, FV best deals with moderate scale and translation changes.

1 Introduction

Recently, there has been an increasing interest on action recognition using Riemannian manifolds. Such recognition systems can be roughly placed into two main categories: **(i)** based on linear subspaces (LS), and **(ii)** based on symmetric positive definite (SPD) matrices. The space of m-dimensional LS in $\mathbb{R}^n$ can be viewed as a special case of Riemannian manifolds, known as Grassmann manifolds [47].

Other techniques have been also applied for the action recognition problem. Among them we can find Gaussian mixture models (GMMs), bag-of-features (BoF), and Fisher vectors (FVs). In [10,27] each action is represented by a combination of GMMs and then the decision making is based on the principle of selecting the most probable action according to Bayes' theorem [7]. The FV representation can be thought as an evolution of the BoF representation, encoding additional information [12]. Rather than encoding the frequency of the descriptors for a given video, FV encodes the deviations from a probabilistic version of the visual dictionary (which is typically a GMM) [9,50].

Several review papers have compared various techniques for human action recognition [1,26,36,52,22]. The reviews show how this research area has progressed throughout the years, discuss the current advantages and limitations of the state-of-the-art, and

---

provide potential directions for addressing the limitations. However, none of them focus on how well various action recognition systems work across same datasets *and* same extracted features. An earlier comparison of classifiers for human activity recognition is studied [34]. The performance comparison with seven classifiers in one single dataset is reported. Although this work presents a broad range of classifiers, it fails to provide a more extensive comparison by using more datasets and hence its conclusions may not generalise to other datasets.

So far there has been no systematic comparison of performance between methods based on SPD matrices and LS. Furthermore, there has been no comparison of manifold based methods against traditional action recognition methods based on GMMs and FVs in the presence of realistic and challenging conditions. Lastly, existing review papers fail to compare various classifiers using the same features across several datasets.

Contributions. To address the aforementioned problems, in this work we provide a more detailed analysis of the performance of the aforementioned methods under the same set of features. To this end, we test with three popular datasets: KTH [41], UCF-Sports [37] and UT-Tower [11]. For the Riemannian representations we use nearest-neighbour classifiers, kernels as well as recent kernelised sparse representations. Finally, we quantitatively show when these methods break and how the performance degrades when the datasets have challenging conditions (translations and scale variations). For a fair comparison across all approaches, we will use the same set of features, as explained in Section 3. More specifically, a video is represented as a set of features extracted on a pixel basis. We also describe how we use this set of features to obtain both Riemannian features: (1) covariance features that lie in the space of SPD matrices, and (2) LS that lie in the space of Grassmann manifolds. In Sections 4 and 5, we summarise learning methods based Riemannian manifolds as well as GMMs and FVs. The datasets and experiment setup are described in Section 6. In Section 7, we present comparative results on three datasets under ideal and challenging conditions. The main findings are summarised in Section 8.

2 Related Work

Many computer vision applications often lie in non-Euclidean spaces, where the underlying distance metric is not the usual l_2 norm [48,24]. For instance, SPD matrices and LS of the Euclidean space are known to lie on Riemannian manifolds. Non-singular covariance matrices are naturally SPD [3] and have been used to describe gesture and action recognition in [13,14,16,40].

Grassmann manifolds, which are special cases of Riemannian manifolds, represent a set of m-dimensional linear subspaces and have also been investigated for the action recognition problem [28,29,30,32]. The straightforward way to deal with Riemannian manifolds is via the nearest-neighbour (NN) scheme. For SPD matrices, NN classification using the log-Euclidean metric for covariance matrices is employed in [46,16]. Canonical or principal angles are used as a metric to measure similarity between two LS and have been employed in conjunction with NN in [46].

Manifolds can be also mapped to a reproducing kernel Hilbert space (RKHS) by using kernels. Kernel analysis on SPD matrices and LS has been used for gesture and action recognition in [19,25,48]. SPD matrices are embedded into RKHS via a pseudo

kernel in [19]. With this pseudo kernel is possible to formulate a locality preserving projections over SPD matrices. Positive definite radial kernels are used to solve the action recognition problem in [25], where an optimisation algorithm is employed to select the best kernel among the class of positive definite radial kernels on the manifold.

Recently, the traditional sparse representation (SR) on vectors has been generalised to sparse representations in SPD matrices and LS [15,20,18,49]. While the objective of SR is to find a representation that efficiently approximates elements of a signal class with as few atoms as possible, for the Riemannian SR, any given point can be represented as a sparse combination of dictionary elements [20,18]. In [18], LS are embedded into the space via isometric mapping, which leads to a closed-form solution for updating a LS representation, atom by atom. Moreover, [18] presents a kernelised version of the dictionary learning algorithm to deal with non-linearity in data. [20] outlines the sparse coding and dictionary learning problem for SPD matrices. To this end, SPD matrices are embedded into the RKHS to perform sparse coding.

GMMs have also been explored for the action detection and classification problems. Each action is represented by a combination of GMMs in [27]. Each action is modelled by two sets of feature attributes. The first set represents the change of body size, while the second represents the speed of the action. Features with high correlations for describing actions are grouped into the same Category Feature Vector (CFV). All CFVs related to the same category are then modelled using a GMM. Other approaches for action recognition using GMM are presented in [8,10]. In [8], spatio-temporal interest points detectors are used to collect a set of local feature vectors, and each feature vector is modelled via GMMs. Based on GMMs, the likelihood of each feature vector belonging to a given action of interests can be estimated. Actions are modelled using a GMM using low-dimensional action features in [10]. For GMMs, the decision making is based on the principle of selecting the most probable action according to Bayes' theorem [44].

Recently, the FV approach has been successfully applied to the action recognition problem [9,33,50]. This approach can be thought as an evolution of the BoF representation, encoding additional information [12,50]. Rather than encoding the frequency of the descriptors, as for BoF, FV encodes the deviations from a probabilistic version of the visual dictionary. This is done by computing the gradient of the sample log-likelihood with respect the parameters of the dictionary model. Since more information is extracted, a smaller visual dictionary size can be used than for BoF, in order to achieve the same or better performance.

3 Video Descriptors

Here, we describe how to extract from a video a set of features on a pixel level. The video descriptor is the same for all the methods examined in this work. A video $\mathcal{V} = \{\boldsymbol{I}_t\}_{t=1}^{T}$ is an ordered set of T frames. Each frame $\boldsymbol{I}_t \in \mathbb{R}^{r \times c}$ can be represented by a set of feature vectors $F_t = \{\boldsymbol{f}_p\}_{p=1}^{N_t}$. We extract the following $d = 14$ dimensional feature vector for each pixel in a given frame t [10]:

$$\boldsymbol{f} = [\, x, \; y, \; \boldsymbol{g}, \; \boldsymbol{o} \,]^{\top} \tag{1}$$

where x and y are the pixel coordinates, while $\boldsymbol{g}$ and $\boldsymbol{o}$ are defined as:

$$\boldsymbol{g} = \left[\ |J_x|, \ |J_y|, \ |J_{yy}|, \ |J_{xx}|, \ \sqrt{J_x^2 + J_y^2}, \ \mathrm{atan}\frac{|J_y|}{|J_x|} \ \right] \tag{2}$$

$$\boldsymbol{o} = \left[\ u, \ v, \ \frac{\partial u}{\partial t}, \ \frac{\partial v}{\partial t}, \ \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right), \ \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right) \ \right] \tag{3}$$

The first four gradient-based features in (2) represent the first and second order intensity gradients at pixel location (x, y). The last two gradient features represent gradient magnitude and gradient orientation. The optical flow based features in (3) represent: the horizontal and vertical components of the flow vector, the first order derivatives with respect to time, the divergence and vorticity of optical flow [4], respectively. Typically only a subset of the pixels in a frame correspond to the object of interest ($N_t < r \times c$). As such, we are only interested in pixels with a gradient magnitude greater than a threshold τ [16]. We discard feature vectors from locations with a small magnitude, resulting in a variable number of feature vectors per frame.

For each video $\mathcal{V}$, the feature vectors are pooled into set $\mathcal{F} = \{\boldsymbol{f}_n\}_{n=1}^N$ containing N vectors. This pooled set of features $\mathcal{F}$ can be used directly by methods such as GMMs and FVs. Describing these features using a Riemannian Manifold setting requires a further step to produce either a covariance matrix feature or a linear subspace feature.

Covariance matrices of features have proved very effective for action recognition [16,40]. The empirical estimate of the covariance matrix of set $\mathcal{F}$ is given by:

$$\boldsymbol{C} = \frac{1}{N} \sum\nolimits_{n=1}^N \left(\boldsymbol{f}_n - \overline{\mathcal{F}}\right) \left(\boldsymbol{f}_n - \overline{\mathcal{F}}\right)^\top \tag{4}$$

where $\overline{\mathcal{F}} = \frac{1}{N} \sum_{n=1}^N \boldsymbol{f}_n$ is the mean feature vector.

The pooled feature vectors set $\mathcal{F}$ can be represented as a linear subspace through any orthogonalisation procedure like singular value decomposition (SVD) [21]. Let $\mathcal{F} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$ be the SVD of $\mathcal{F}$. The first m columns of $\boldsymbol{U}$ represent an optimised subspace of order m. The Grassmann manifold $\mathcal{G}_{d,m}$ is the set of m-dimensional linear subspaces of $\mathbb{R}^d$. An element of $\mathcal{G}_{d,m}$ can be represented by an orthonormal matrix $\boldsymbol{Y}$ of size $d \times m$ such that $\boldsymbol{Y}^\top \boldsymbol{Y} = \boldsymbol{I}_m$, where $\boldsymbol{I}_m$ is the $m \times m$ identity matrix.

4 Classification on Riemannian Manifolds

4.1 Nearest-Neighbour Classifier

The Nearest Neighbour (NN) approach classifies a query data based on the most similar observation in the annotated training set [31]. To decide whether two observations are similar we will employ two metrics: the log-Euclidean distance for SPD matrices [16] and the Projection Metric for LS [17].

The log-Euclidean distance (d_{spd}) is one of the most popular metrics for SPD matrices due to its accuracy and low computational complexity [5]; it is defined as:

$$d_{\mathrm{spd}}(\boldsymbol{C}_1, \boldsymbol{C}_2) = || \log(\boldsymbol{C}_1) - \log(\boldsymbol{C}_2) ||_F \tag{5}$$

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

As for LS, a common metric to measure the similarity between two subspaces is via principal angles [17]. The metric can include the smallest principal angle, the largest principal angle, or a combination of all principal angles [17,48]. In this work we have selected the Projection Metric which uses all the principal angles [17]:

$$d_{\text{ls}}(\boldsymbol{Y}_1, \boldsymbol{Y}_2) = \left(m - \sum\nolimits_{i=1}^{m} \cos^2 \theta_i \right)^{1/2} \tag{6}$$

where m is the size of the subspace.

The principal angles can be easily computed from the SVD of $\boldsymbol{Y}_1^\top \boldsymbol{Y}_2 = \boldsymbol{U}(\cos \Theta)\boldsymbol{V}^\top$, where $\boldsymbol{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_m]$, $\boldsymbol{V} = [\boldsymbol{v}_1 \cdots \boldsymbol{v}_m]$, and $\cos \Theta = \text{diag}(\cos \theta_1, \cdots, \cos \theta_m)$.

4.2 Kernel Approach

Manifolds can be mapped to Euclidean spaces using Mercer kernels [48,20,24]. This transformation allows us to employ algorithms originally formulated for $\mathbb{R}^n$ with manifold value data. Several kernels for the set of SPD matrices have been proposed in the literature [20,24,54]. One kernel based on the log-Euclidean distance is derived in [51] and various kernels can be generated, including [48]:

$$K_{\text{spd}}^{\text{rbf}}(\boldsymbol{C}_1, \boldsymbol{C}_2) = \exp\left(-\gamma_r \cdot || \log(\boldsymbol{C}_1) - \log(\boldsymbol{C}_2)||_F^2\right) \tag{7}$$

$$K_{\text{spd}}^{\text{poly}}(\boldsymbol{C}_1, \boldsymbol{C}_2) = \left(\gamma_p \cdot \text{tr}\left[\log(\boldsymbol{C}_1)^\top \log(\boldsymbol{C}_2)\right]\right)^d \tag{8}$$

Similar to SPD kernels, many kernels have been proposed for LS [21,42]. Various kernels can be generated from the projection metric, such as [48]:

$$K_{\text{ls}}^{\text{rbf}}(\boldsymbol{Y}_1, \boldsymbol{Y}_2) = \exp\left(-\gamma_r \cdot ||\boldsymbol{Y}_1 \boldsymbol{Y}_1^\top - \boldsymbol{Y}_2 \boldsymbol{Y}_2^\top||_F^2\right) \tag{9}$$

$$K_{\text{ls}}^{\text{poly}}(\boldsymbol{Y}_1, \boldsymbol{Y}_2) = \left(\gamma_p \cdot ||\boldsymbol{Y}_1^\top \boldsymbol{Y}_2||_F^2\right)^m \tag{10}$$

The parameters γ_r and γ_p are defined in Section 6. The kernels are used in combination with Support Vector Machines (SVMs) [7].

4.3 Kernelised Sparse Representation

Recently, several works show the efficacy of sparse representation methods for addressing manifold feature classification problems [53,18]. Here, each manifold point is represented by its sparse coefficients. Let $\mathcal{X} = \{\boldsymbol{X}_j\}_{j=1}^{J}$ be a population of Riemannian points (where $\boldsymbol{X}_j$ is either a SPD matrix or a LS) and $\mathcal{D} = \{\boldsymbol{D}_i\}_{i=1}^{K}$ be the Riemannian dictionary of size K, where each element represents an atom. Given a kernel $k(\cdot, \cdot)$, induced by the feature mapping function $\phi : \mathbb{R}^d \to \mathbb{H}$, we seek to learn a dictionary and corresponding sparse code $\boldsymbol{s} \in \mathbb{R}^K$ such that $\phi(\boldsymbol{X})$ can be well approximated by the dictionary $\phi(\mathcal{D})$. The kernelised dictionary learning in Riemannian manifolds optimises the following objective function [53,18]:

$$\min_{\boldsymbol{s}} \left(\left\| \phi(\boldsymbol{X}) - \sum\nolimits_{i=1}^{K} s_i \phi(\boldsymbol{D}_i) \right\|_F^2 + \lambda \left\| \boldsymbol{s} \right\|_1 \right) \qquad (11)$$

over the dictionary and the sparse codes $\mathcal{S} = \{\boldsymbol{s}_j\}_{j=1}^J$. After initialising the dictionary $\mathcal{D}$, the objective function is solved by repeating two steps (sparse coding and dictionary update). In the sparse coding step, $\mathcal{D}$ is fixed and $\mathcal{S}$ is computed. In the dictionary update step, $\mathcal{S}$ is fixed while $\mathcal{D}$ is updated, with each dictionary atom updated independently.

For the sparse representation on SPD matrices, each atom $\boldsymbol{D}_i \in \mathbb{R}^{d \times d}$ and each element $\boldsymbol{X} \in \mathbb{R}^{d \times d}$ are SPD matrices. The dictionary is learned following [20], where the dictionary is initialised using the Karcher mean [6]. For the sparse representation on LS, the dictionary $\boldsymbol{D}_i \in \mathbb{R}^{d \times m}$ and each element $\boldsymbol{X} \in \mathbb{R}^{d \times m}$ are elements of $\mathcal{G}_{d,m}$ and need to be determined by the Kernelised Grassmann Dictionary Learning algorithm proposed in [18]. We refer to the kernelised sparse representation (KSR) for SPD matrices and LS as KSR$_\text{spd}$ and KSR$_\text{ls}$, respectively.

5 Classification via Gaussian Mixture Models and Fisher Vectors

A **Gaussian Mixture Model** (GMM) is a weighted sum of K component Gaussian densities [7]:

$$p(\boldsymbol{f}|\lambda) = \sum\nolimits_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad (12)$$

where $\boldsymbol{f}$ is a d-dimensional feature vector, w_k is the weight of the k-th Gaussian (with constraints $0 \leq w_k \leq 1$ and $\sum_{k=1}^{K} w_k = 1$), and $\mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the component Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, given by:

$$\mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(\boldsymbol{f} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{f} - \boldsymbol{\mu}) \right\} \qquad (13)$$

The complete Gaussian mixture model is parameterised by the mean vectors, covariance matrices and weights of all component densities. These parameters are collectively represented by the notation $\lambda = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$. For the GMM, we learn one model per action. This results in a set of GMM models that we will express as $\{\lambda_a\}_{a=1}^A$, where A is the total number of actions. For each testing video $\mathcal{V}$, the feature vectors in set $\mathcal{F}$ are assumed independent, so the average log-likelihood of a model λ_a is computed as:

$$\log p(\mathcal{F}|\lambda_a) = \frac{1}{N} \sum\nolimits_{n=1}^{N} \log p(\boldsymbol{f}_n|\lambda_a) \qquad (14)$$

We classify each video to the model a which has the highest average log-likelihood.

The **Fisher Vector** (FV) approach encodes the deviations from a probabilistic visual dictionary, which is typically a GMM with diagonal covariance matrices [38]. The parameters of a GMM with K components can be expressed as $\lambda = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k\}_{k=1}^K$, where, w_k is the weight, $\boldsymbol{\mu}_k$ is the mean vector, and $\boldsymbol{\sigma}_k$ is the diagonal covariance matrix for the k-th Gaussian. The parameters are learned using the Expectation Maximisation algorithm [7] on training data. Given the pooled set of features $\mathcal{F}$ from video $\mathcal{V}$, the deviations from the GMM are then accumulated using [38]:

$$\mathcal{G}^{\mathcal{F}}_{\boldsymbol{\mu}_k} = \frac{1}{N\sqrt{w_k}} \sum_{n=1}^{N} \gamma_n(k) \left(\frac{\boldsymbol{f}_n - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right) \qquad (15)$$

$$\mathcal{G}^{\mathcal{F}}_{\boldsymbol{\sigma}_k} = \frac{1}{N\sqrt{2w_k}} \sum_{n=1}^{N} \gamma_n(k) \left[\frac{(\boldsymbol{f}_n - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1 \right] \qquad (16)$$

where vector division indicates element-wise division and $\gamma_n(k)$ is the posterior probability of $\boldsymbol{f}_n$ for the k-th component:

$$\gamma_n(k) = \frac{w_k \mathcal{N}(\boldsymbol{f}_n | \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)}{\sum_{i=1}^{K} w_i \mathcal{N}(\boldsymbol{f}_n | \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)} \qquad (17)$$

The Fisher vector for each video $\mathcal{V}$ is represented as the concatenation of $\mathcal{G}^{\mathcal{F}}_{\boldsymbol{\mu}_k}$ and $\mathcal{G}^{\mathcal{F}}_{\boldsymbol{\sigma}_k}$ (for $k = 1, \ldots, K$) into vector $\mathcal{G}^{\mathcal{F}}_{\lambda}$. As $\mathcal{G}^{\mathcal{F}}_{\boldsymbol{\mu}_k}$ and $\mathcal{G}^{\mathcal{F}}_{\boldsymbol{\sigma}_k}$ are d-dimensional, $\mathcal{G}^{\mathcal{F}}_{\lambda}$ has the dimensionality of $2dK$. Power normalisation is then applied to each dimension in $\mathcal{G}^{\mathcal{F}}_{\lambda}$. The power normalisation to improve the FV for classification was proposed in [35] of the form $z \leftarrow \text{sign}(z)|z|^{\rho}$, where z corresponds to each dimension and the power coefficient $\rho = 1/2$. Finally, l_2-normalisation is applied. Note that we have omitted the deviations for the weights as they add little information [38]. The FVs are fed to a linear SVM for classification, where the similarity between vectors is measured using dot-products [38].

6 Datasets and Setup

For our experiments, we use three datasets: KTH [41], UCF-Sports [37], and UT-Tower [11]. See Fig. 1 for examples of actions. In the following sections, we describe how each of the datasets is employed, and we provide a description of the setup used for the experiments.

Datasets. The KTH dataset [41] contains 25 subjects performing 6 types of human actions and 4 scenarios. The actions included in this dataset are: boxing, handclapping, handwaving, jogging, running, and walking. The scenarios include indoor, outdoor, scale variations, and varying clothes. Each original video of the KTH dataset contains an individual performing the same action. The image size is 160×120 pixels, and temporal resolution is 25 frames per second. For our experiments we only use scenario 1.

The UCF-Sports dataset [37] is a collections of 150 sport videos or sequences. This datasets consists of 10 actions: diving, golf swinging, kicking a ball, lifting weights, riding horse, running, skate boarding, pommel horse, high bar, and walking. The number of videos per action varies from 6 to 22. The videos presented in this dataset have varying backgrounds. We use the bounding box enclosing the person of interest provided with the dataset where available. We create the corresponding 10 bounding boxes not provided with the dataset. The bounding box size is 250×400 pixels.

The UT-Tower dataset [11] contains 9 actions performed 12 times. In total, there are 108 low-resolution videos. The actions include: pointing, standing, digging, walking, carrying, running, wave1, wave2, and jumping. The videos were recorded in two scenes (concrete square and lawn). As the provided bounding boxes have variable sizes, we resize the resulting boxes to 32×32.

Fig. 1. Examples from the three datasets.

Setup. We use the Leave-One-Out (LOO) protocol suggested by each dataset. We leave one sample video out for testing on a rotating basis for UT-Tower and UCF-Sports. For KTH we leave one person out. For each video we extract a set of $d = 14$ dimensional features vectors as explained in Section 3. We only use feature vectors with a gradient magnitude greater that a threshold τ. The threshold τ used for selecting low-level feature vectors was set to 40 as per [10].

For each video, we obtain one SPD matrix and one LS. In order to obtain the optimised linear subspace $\mathcal{G}_{d,m}$ in the the manifold representation, we vary $m = 1, \cdots, d$. We test with manifolds kernels using various parameters. The set of parameters was used as proposed in [48]. Polynomial kernels $K_{\text{spd}}^{\text{poly}}$ and $K_{\text{ls}}^{\text{rbf}}$ are generated by taking $\gamma_p = 1/d_p$ and $d_p = \{1, 2, \cdots, d\}$. Projection RBF kernels are generated with $\gamma_r = \frac{1}{d} 2^{\delta}$ and $\delta = \{-10, -9, \cdots, 9\}$ for $K_{\text{spd}}^{\text{rbf}}$, and $\delta = \{-14, -12, \cdots, 20\}$ for $K_{\text{ls}}^{\text{rbf}}$. For the sparse representation of SPD matrices and LS we have used the code provided by [20,18]. Kernels are used in combination with SVM for final classification. We report the best accuracy performance after iterating with various parameters.

For the FV representation, we use the same set-up as in [50]. We randomly sampled 256,000 features from training videos and then the visual dictionary is learned with 256 Gaussians. Each video is represented by a FV. The FVs are fed to a linear SVM for classification. For the GMM modelling, we learn a model for each action using all the feature vectors belonging to the same action. For each action a GMM is trained with $K = 256$ components. The experiments were implemented with the aid of the Armadillo C++ library [39].

7 Comparative Evaluation

We perform two sets of experiments: **(i)** in ideal conditions, where the classification is carried out using each original dataset, and **(ii)** in realistic and challenging conditions where testing videos are modified by scale changes and translations.

7.1 Ideal Conditions

We start our experiments using the NN classifier for both Riemannian representations: SPD matrices and LS. For LS we employ the projection metric as per Eq. (6) and for

Table 1. Accuracy of action recognition in ideal conditions.

	KTH	UCF-Sports	UT-Tower	average
d_{spd} + NN	76.0%	76.5%	73.1%	75.2%
d_{ls} + NN	67.3%	65.7%	76.8%	69.9%
$K_{\mathrm{spd}}^{\mathrm{poly}}$ + SVM	92.0%	75.2%	87.9%	<u>85.0%</u>
$K_{\mathrm{spd}}^{\mathrm{rbf}}$ + SVM	84.0%	79.2%	81.5%	81.6%
$K_{\mathrm{ls}}^{\mathrm{poly}}$ + SVM	56.0%	50.3%	42.6%	49.6%
$K_{\mathrm{ls}}^{\mathrm{rbf}}$ + SVM	76.0%	61.7%	79.6%	72.4%
KSR$_{\mathrm{spd}}$ + SVM	80.0%	76.5%	81.5%	79.3%
KSR$_{\mathrm{ls}}$ + SVM	74.0%	72.5%	83.3%	77.3%
GMM	86.7%	80.5%	87.9%	<u>85.0%</u>
FV + SVM	**96.7**%	**88.6**%	**92.5**%	**92.6**%

SPD matrices we employ the log-Euclidean distance as per Eq. (5). We tune the parameter m (subspace order) for each dataset. The kernels selected for SPD matrices and LS are described in Eqs. (7)-(10) and their parameters are selected as per Section 6.

We present a summary of the best performance obtained for the manifold representations using the optimal subspace for LS and also the optimal kernel parameters for both representations. Similarly, we report the best accuracy performance for the kernelised sparse representations KSR$_{\mathrm{spd}}$ and KSR$_{\mathrm{ls}}$. Moreover, we include the performance for the GMM and FV representations.

The results are presented in Table 1. First of all, we observe that using a SVM for action recognition usually leads to a better accuracy than NN. In particular, we notice that the NN approach performs quite poorly. The NN classifier may not be effective enough to capture the complexity of the human actions when there is insufficient representation of the actions (one video is represented by one SPD matrix or one LS). Secondly, we observe that among the manifold techniques, SPD based approaches perform better than LS based approaches. While LS capture only the dominant eigenvectors [43], SPD matrices capture both the eigenvectors and eigenvalues [45]. The eigenvalues of a covariance matrix typify the variance captured in the direction of each eigenvector [45].

Despite KSR$_{\mathrm{spd}}$ showing superior performance in other computer vision tasks [20], it is not the case for the action recognition problem. We conjecture this is due to the lack of labelled training data (each video is represented by only one SPD matrix), which may yield a dictionary with bad generalisation power. Moreover, sparse representations can be over-pruned, being caused by discarding several representative points that may be potentially useful for prediction [23].

Although kernel approaches map the data into higher spaces to allow linear separability, K_{spd}^{poly} exhibits on average a similar accuracy to GMM which does not transform the data. GMM is a weighted sum of Gaussian probability densities, which in addition to the covariance matrices, it uses the means and weights to determine the average log-likelihood of a set of feature vectors from a video to belong to a specific action. While

SPD kernels only use covariance matrices, GMMs use both covariance matrices and means. The combination of both statistics has proved to increase the accuracy performance in other classification tasks [2,38]. FV outperforms all the classification methods with an average accuracy of 92.6%, which is 7.6 points higher than both GMM and K_{spd}^{poly}. Similarly to GMM, FV also incorporates first and second order statistics (means and covariances), but it has additional processing in the form of power normalisation. It is shown in [35] that when the number of Gaussians increases, the FV turns into a sparser representation and it negatively affects the linear SVM which measures the similarity using dot-products. The power normalisation unsparsifies the FV making it more suitable for linear SVMs. The additional information provided by the means and the power normalisation explains the superior accuracy performance of FV.

7.2 Challenging Conditions

In this section, we evaluate the performance on all datasets under consideration when the testing videos have translations and scale variations. We have selected the following approaches for this evaluation: d_{spd}, $K_{\text{spd}}^{\text{poly}}$, $K_{\text{ls}}^{\text{rbf}}$, and FV. We discard d_{ls}, as its performance is too low and presents similar behaviour to d_{spd}. We do not include experiments on KSR_{spd} and KSR_{ls}, as we found that they show similar trends as $K_{\text{spd}}^{\text{poly}}$ and $K_{\text{ls}}^{\text{rbf}}$, respectively. Alike, GMM exhibits similar behaviour as FV.

For this set of experiments, the training is carried out using the original datasets. For the analysis of translations, we have translated (shifted) each testing video vertically and horizontally. For the evaluation under scale variations, each testing video is shrunk or magnified. For both cases, we replace the missing pixels simply by copying the nearest rows or columns. See Fig. 2 for examples of videos under challenging conditions.

The results for scale variations and translations are shown in Figs. 3 and 4, respectively. These results reveal that all the analysed approaches are susceptible to translation and scale variations. Both kernel based methods, $K_{\text{spd}}^{\text{poly}}$ and $K_{\text{ls}}^{\text{rbf}}$, exhibit sharp performance degradation even when the scale is only magnified or compressed by a factor of 0.05%. Similarly, for both kernels the accuracy rapidly decreases with a small translation. The NN classification using the log-Euclidean distance (d_{spd}) is less sensitive to both variations. It can be explained by the fact that log-Euclidean metrics are by definition invariant by any translation and scaling in the domain of logarithms [5]. FV presents the best behaviour under moderate variations in both scale and translation. We attribute this to the loss of explicit spatial relations between object parts.
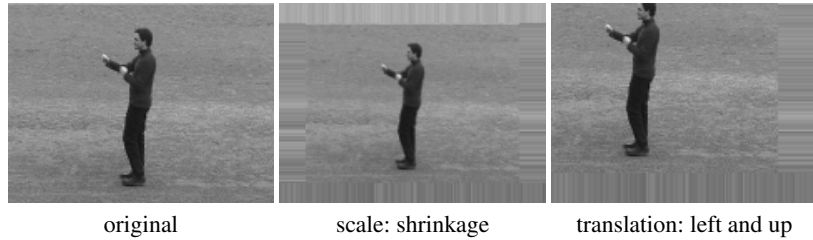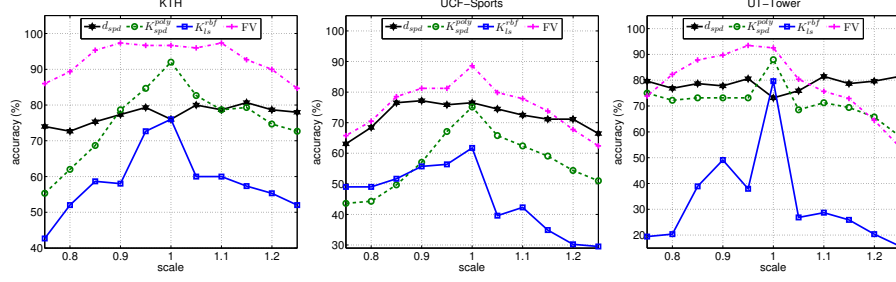

| original | scale: shrinkage | translation: left and up |

Fig. 2. Examples of challenging conditions.

Fig. 3. Results for scale variation; scale > 1 means magnification, while < 1 means shrinkage.
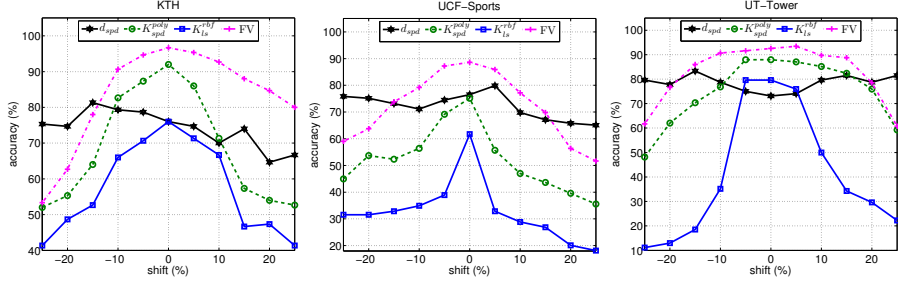

Fig. 4. Results for translation experiments. Each testing video is translated vertically and horizontally at the same time. A positive percentage indicates the video has been translated to the right and bottom while a negative percentage indicates the video has been translated to the left and up.

8   Main Findings

In this paper, we have presented an extensive empirical comparison among existing techniques for the human action recognition problem. We have carried out our experiments using three popular datasets: KTH, UCF-Sports and UT-Tower. We have analysed Riemannian representations including nearest-neighbour classification, kernel methods, and kernelised sparse representations. For Riemannian representation we used covariance matrices of features, which are symmetric positive definite (SPD), as well as linear subspaces (LS). Moreover, we compared all the aforementioned Riemannian representations with GMM and FV based representations, using the same extracted features. We also evaluated the robustness of the most representative approaches to translation and scale variations.

For manifold representations, all SPD matrices approaches surpass their LS counterpart, as a result of the use of not only the dominant eigenvectors but also the eigenvalues. The FV representation outperforms all the techniques under ideal and challenging conditions. Under ideal conditions, FV achieves an overall accuracy of 92.6%, which is 7.6 points higher than both GMM and the polynomial kernel using SPD matrices (K_{spd}^{poly}). FV encodes more information than Riemmannian based methods, as it characterises the deviation from a probabilistic visual dictionary (a GMM) using means and covariance matrices. Moreover, FV is less sensitive under moderate variations in both scale and translation.

References

1. J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Comput. Surveys*, 43(3):16:1–16:43, 2011.
2. N. Aggarwal and R. Agrawal. First and second order statistics features for classification of magnetic resonance brain images. *Journal of Signal and Information Processing*, 3:146–153, 2012.
3. A. Alavi, A. Wiliem, K. Zhao, B. Lovell, and C. Sanderson. Random projections on manifolds of symmetric positive definite matrices for image classification. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 301–308, 2014.
4. S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *Pattern Analysis and Machine Intelligence*, 32(2):288–303, 2010.
5. V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. In *Magnetic Resonance in Medicine*, 2006.
6. D. A. Bini and B. Iannazzo. Computing the Karcher mean of symmetric positive definite matrices. *Linear Algebra and its Applications*, 438(4):1700 – 1710, 2013.
7. C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
8. L. Cao, Y. Tian, Z. Liu, B. Yao, Z. Zhang, and T. Huang. Action detection using multiple spatial-temporal interest point features. In *International Conference on Multimedia and Expo (ICME)*, pages 340–345, 2010.
9. J. Carvajal, C. McCool, B. Lovell, and C. Sanderson. Joint recognition and segmentation of actions via probabilistic integration of spatio-temporal Fisher vectors. In *Lecture Notes in Artificial Intelligence (LNAI), Vol. 9794*, pages 115–127, 2016.
10. J. Carvajal, C. Sanderson, C. McCool, and B. C. Lovell. Multi-action recognition via stochastic modelling of optical flow and gradients. In *Workshop on Machine Learning for Sensory Data Analysis (MLSDA)*, pages 19–24, 2014.
11. C.-C. Chen, M. S. Ryoo, and J. K. Aggarwal. UT-Tower Dataset: Aerial View Activity Classification Challenge, 2010.
12. G. Csurka and F. Perronnin. Fisher vectors: Beyond bag-of-visual-words image representations. In *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, volume 229, pages 28–42. 2011.
13. M. Faraki, M. T. Harandi, and F. Porikli. More about VLAD: A leap from Euclidean to Riemannian manifolds. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
14. M. Faraki, M. Palhang, and C. Sanderson. Log-Euclidean bag of words for human action recognition. *IET Computer Vision*, 9(3):331–339, 2015.
15. K. Guo, P. Ishwar, and J. Konrad. Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels. In *Recognizing Patterns in Signals, Speech, Images and Videos*, volume 6388 of *Lecture Notes in Computer Science*, pages 294–305. Springer Berlin Heidelberg, 2010.
16. K. Guo, P. Ishwar, and J. Konrad. Action recognition from video using feature covariance matrices. *IEEE Transactions on Image Processing*, 22(6):2479–2494, 2013.
17. J. Hamm and D. D. Lee. Grassmann discriminant analysis: A unifying view on subspace-based learning. In *International Conference on Machine Learning (ICML)*, pages 376–383, 2008.

18. M. Harandi, C. Sanderson, C. Shen, and B. Lovell. Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution. In *International Conference on Computer Vision (ICCV)*, pages 3120–3127, 2013.

19. M. Harandi, C. Sanderson, A. Wiliem, and B. Lovell. Kernel analysis over Riemannian manifolds for visual recognition of actions, pedestrians and textures. In *Workshop on Applications of Computer Vision (WACV)*, pages 433–439, 2012.

20. M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science (LNCS)*, volume 7573, pages 216–229. 2012.

21. M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell. Kernel analysis on Grassmann manifolds for action recognition. *Pattern Recognition Letters*, 34(15):1906–1915, 2013.

22. T. Hassner. A critical review of action recognition benchmarks. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 245–250, 2013.

23. S. Hirose, I. Nambu, and E. Naito. An empirical solution for over-pruning with a novel ensemble-learning method for fMRI decoding. *Journal of Neuroscience Methods*, 239:238–245, 2015.

24. S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on the Riemannian manifold of symmetric positive definite matrices. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 73–80, 2013.

25. S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Optimizing over radial kernels on compact manifolds. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3802–3809, 2014.

26. S.-R. Ke, H. L. U. Thuc, Y.-J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi. A review on video-based human activity recognition. *Computers*, 2(2):88, 2013.

27. W. Lin, M.-T. Sun, R. Poovandran, and Z. Zhang. Human activity recognition for video surveillance. In *International Symposium on Circuits and Systems (ISCAS)*, pages 2737–2740, 2008.

28. Y. M. Lui. Human gesture recognition on product manifolds. *Journal of Machine Learning Research*, 13(1):3297–3321, 2012.

29. Y. M. Lui. Tangent bundles on special manifolds for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(6):930–942, 2012.

30. Y. M. Lui and J. Beveridge. Tangent bundle for human action recognition. In *International Conference on Automatic Face Gesture Recognition and Workshops*, pages 97–102, 2011.

31. M. N. Murty and V. S. Devi. Nearest neighbour based classifiers. In *Pattern Recognition*, Undergraduate Topics in Computer Science, pages 48–85. Springer London, 2011.

32. S. O'Hara and B. Draper. Scalable action recognition with a subspace forest. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1210–1217, June 2012.

33. D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with Fisher vectors on a compact feature set. In *International Conference on Computer Vision (ICCV)*, pages 1817–1824, 2013.

34. Ó. Pérez, M. Piccardi, J. García, and J. M. Molina. Comparison of classifiers for human activity recognition. In *Nature Inspired Problem-Solving Methods in Knowledge Engineering*, volume 4528 of *Lecture Notes in Computer Science*, pages 192–201. 2007.

35. F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *European Conference on Computer Vision (ECCV)*, volume 6314, pages 143–156. 2010.

36. R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976 – 990, 2010.

37. M. Rodriguez, J. Ahmed, and M. Shah. Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

38. J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the Fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

39. C. Sanderson and R. Curtin. Armadillo: a template-based C++ library for linear algebra. *Journal of Open Source Software*, 1:26, 2016.

40. A. Sanin, C. Sanderson, M. Harandi, and B. Lovell. Spatio-temporal covariance descriptors for action and gesture recognition. In *Workshop on Applications of Computer Vision (WACV)*, pages 103–110, 2013.

41. C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *International Conference on Pattern Recognition (ICPR)*, volume 3, pages 32–36, 2004.

42. S. Shirazi, M. Harandi, C. Sanderson, A. Alavi, and B. Lovell. Clustering on Grassmann manifolds via kernel embedding with application to action analysis. In *International Conference on Image Processing (ICIP)*, pages 781–784, 2012.

43. S. Shirazi, C. Sanderson, C. McCool, and M. T. Harandi. Bags of affine subspaces for robust object tracking. In *International Conference on Digital Image Computing: Techniques and Applications*, 2015.

44. S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, Boston, 4th edition, 2009.

45. I. Traore and A. A. E. Ahmed. *Continuous Authentication Using Biometrics: Data, Models, and Metrics*. IGI Global, Hershey, PA, USA, 1st edition, 2011.

46. P. Turaga and R. Chellappa. Nearest-neighbor search algorithms on non-Euclidean manifolds for computer vision applications. In *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, pages 282–289. ACM, 2010.

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*, 33(11):2273–2286, 2011.

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

49. B. Wang, Y. Hu, J. Gao, Y. Sun, and B. Yin. Low rank representation on Grassmann manifolds. In D. Cremers, I. Reid, H. Saito, and M.-H. Yang, editors, *Computer Vision – ACCV 2014*, volume 9003 of *Lecture Notes in Computer Science*, pages 81–96. Springer International Publishing, 2015.

50. H. Wang and C. Schmid. Action recognition with improved trajectories. In *International Conference on Computer Vision (ICCV)*, 2013.

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

52. D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224 – 241, 2011.

53. Y. Wu, Y. Jia, P. Li, J. Zhang, and J. Yuan. Manifold kernel sparse representation of symmetric positive-definite matrices and its applications. *IEEE Transactions on Image Processing*, 24(11):3729–3741, 2015.

54. J. Zhang, L. Wang, L. Zhou, and W. Li. Learning discriminative Stein kernel for SPD matrices and its applications. *IEEE Transactions on Neural Networks and Learning Systems*, in press.