When Kernel Methods meet Feature Learning: Log-Covariance Network for Action Recognition from Skeletal Data

Jacopo Cavazza[1,2], Pietro Morerio[1] and Vittorio Murino[1,3]

[1]Pattern Analysis & Computer Vision (PAVIS) – Istituto Italiano di Tecnologia – *Genova, Italy*
[2]Electrical, Electronics and Telecommunication Engineering and Naval Architecture Department (DITEN) – Università degli Studi di Genova – *Genova, Italy*
[3]Computer Science Department – Università di Verona – *Verona, Italy*
{pietro.morerio,jacopo.cavazza,vittorio.murino}@iit.it

Abstract

Human action recognition from skeletal data is a hot research topic and important in many open domain applications of computer vision, thanks to recently introduced 3D sensors. In the literature, naive methods simply transfer off-the-shelf techniques from video to the skeletal representation. However, the current state-of-the-art is contended between to different paradigms: kernel-based methods and feature learning with (recurrent) neural networks. Both approaches show strong performances, yet they exhibit heavy, but complementary, drawbacks. Motivated by this fact, our work aims at combining together the best of the two paradigms, by proposing an approach where a shallow network is fed with a covariance representation. Our intuition is that, as long as the dynamics is effectively modeled, there is no need for the classification network to be deep nor recurrent in order to score favorably. We validate this hypothesis in a broad experimental analysis over 6 publicly available datasets.

1. Introduction

Human action recognition is a paramount domain in many applicative fields, such as crowd analysis and surveillance, elderly care and autonomous driving vehicles, to name a few.

Despite the wide interest in video-based approaches, this type of data is intrinsically affected by several issues, *e.g.* privacy, occlusions, light variations and background noise. An effective alternative to deal with these challenges is represented by skeletal based representation. This paradigm relies on theoretical guarantees concerning motion perception. It has in fact been proved by Johansson [17] that the displacement of light sources located on keypoints on the humans' skeleton are enough for the visual system to recognizing the displayed action (such as walking, Fig. 1, top).

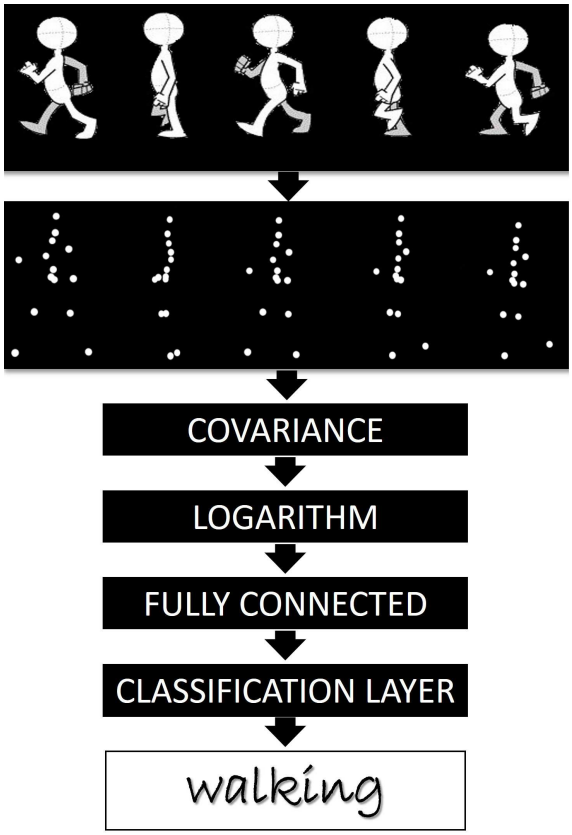

Figure 1. Overview of the proposed *Log-COV-Net*. Starting from a time series of skeletal representations (top), we process the input data (1) with a covariance matrix (2) which is then log-projected (4). A (separately trained) fully connected layer provides the final representation for the classification stage.

Grounding on that, the evolution of systems which can

acquire the skeletal joints nowadays guarantees a reliable estimate of 3D body posture - motion capture, *e.g.* VICON - and a cheap price - depth sensors, *e.g.* Kinect. Additionally, replacing videos with skeletal data does not change the overall general pipeline of action classification: learning/engineering feature representation from trimmed sequences, in order to train a classifier. In practice, for a general action $\mathfrak{a}$, skeletal data is acquired in the form of the following multi-dimensional time-series.

$$\mathbf{P}_{\mathfrak{a}} = \begin{bmatrix} x_1(t=1) & x_1(t=2) & \dots & x_1(t=T) \\ y_1(t=1) & y_1(t=2) & \dots & y_1(t=T) \\ z_1(t=1) & z_1(t=2) & \dots & z_1(t=T) \\ x_2(t=1) & x_2(t=2) & \dots & x_2(t=T) \\ y_2(t=1) & y_2(t=2) & \dots & y_2(t=T) \\ z_2(t=1) & z_2(t=2) & \dots & z_2(t=T) \\ \vdots & \vdots & \vdots & \vdots \\ x_J(t=1) & x_J(t=2) & \dots & x_J(t=T) \\ y_J(t=1) & y_J(t=2) & \dots & y_J(t=T) \\ z_J(t=1) & z_J(t=2) & \dots & z_J(t=T) \end{bmatrix} \quad (1)$$

where columns correspond to timestamps (from $t = 1$ to $t = T$) and triplets of rows $[x_i(t), y_i(t), z_i(t)]^\top$ correspond to 3D spatial coordinates of the i-th joint, $i = 1, \dots, J$.

Related work

In principle, if each frame in a *video* was vectorized, we could gather a data matrix very similar to (1). This is why, especially in the past year, many algorithms, originally devised for video-based action recognition, have been just brought to the skeletal data paradigm upon minor modifications. Among others, we can mention histogram based representations to perform temporal pooling [42, 41, 6], extraction of local spatio-temporal features from the data [4, 30], also applying bag-of-words or Fisher vector approaches to aggregate the raw joint representation in a unique action descriptor [6, 1].

However, the performance of transferring a video-based approach to skeletal data has been proved to be suboptimal with respect to more principled method which endows $\mathbf{P}_{\mathfrak{a}}$ with some kind of structure which can be exploited in the classification stage. We call this structured representation a *kernel*. Among the many proposed kernels, we can recall the representation of each joint trajectory as a roto-translation matrix [35, 36, 15], which leads to exploit the Lie group and Lie algebra properties of the Special Euclidean group. Alternatively, Hankel matrices [22, 43] have been attested to be extremely effective in the field of action recognition from skeletal data, either being paired with Hidden Markov Models [22] or with a prototype-based nearest neighbor classification on the Riemannian manifold [43]. Actually, in both cases of roto-translations and Hankel matrix, countermeasures (such as warping [35, 36]) needs to be taken against the following issue: in (1), while J is fixed (being an intrinsic parameter of the device used for skeleton's acquisition), T is not, and can in fact changes from action to action (and even among repetition of the same action performed by the same person). Therefore, a pre-processing step (such as warping [35, 36]) needs to be applied in order to fix T across instances, since standard methods only deal with fixed-length inputs.

In this respect, the *covariance representation* is a very straightforward workaround. Formally, the covariance matrix $\mathbf{X}_{\mathfrak{a}}$ related to (1) is defined as

$$\mathbf{X}_{\mathfrak{a}} = \frac{1}{T-1} \mathbf{P}_{\mathfrak{a}} \left(\frac{1}{T} \mathbf{I}_T - \mathbf{1}_T \right) \mathbf{P}_{\mathfrak{a}}^\top \quad (2)$$

where $\mathbf{I}_T$ is the identity matrix and $\mathbf{1}_T$ is the $T \times T$ matrix whose all entries are equal to 1. By definition (2), $\mathbf{X}_{\mathfrak{a}}$ is a $3J \times 3J$ SPD (symmetric and positive definite) matrix: we have therefore a fixed-dimensional representation, no matter which is the length T of the time series (1). In fact the index t is saturated by the summations related to the row-by-column matrix products in (2).

In addition to the remarkable property of being invariant with respect to sequence length T, the covariance representation was proved to be an effective tool for action classification [16, 11, 37, 10, 3]. The reason for this lies in the statistical computation of second order temporal momentum of $\mathbf{P}_{\mathfrak{a}}$, the latter being very discriminative in recognizing human actions [16, 37, 3].

Recently, with the introduction of the first big dataset for action recognition with skeletal representation [32], kernel methods are frequently difficult to scale up. This is due the prohibitive dimension of the training/testing Gram matrices, which compute the (kernel) pairwise similarity *for every couple of instances in the dataset*. Thus, they have a quadratic cost as a function of the number of examples. Therefore, their big size make them simply intractable under a computational perspective. In order to circumvent such problem, deep learning is an alternative class of state-of-the-art approaches. In [5, 15, 32, 21] hierarchical feature representations are learnt from the data itself, providing an end-to-end trainable encoding & scalable classification pipeline. However, the reason for their success depends on the big number of free parameters to optimize, being the latter step complicated (the optimization is non-convex, overfitting is a real issue [24]) and computationally intense (GPU acceleration is fundamental).

In light of the dichotomy kernels vs. deep nets, many works have attempted to interconnect the two opposed paradigms. Namely, implementing kernel methods as neural networks (*e.g.* deformable part models [9], multiple kernel learning [29]) or kernelizing existing neural network architectures (*e.g.* convolutional kernel networks [23], SVM neural networks [33]). Recently, neural networks have been tailored to be fed with structured matrices (covariance ma-

trices [14] and rotation matrices [15]). Indeed, classical operations have been re-formulated to accommodate for the different type of input data adopted: for instance, max pooling is performed on the eigenvalues only [33].

With respect to [9, 29, 23, 33] we notice that, in some cases, the connections between the two classes are weak in the sense that one of them impose its own formalism on the other (*e.g.* using backpropagation in multiple kernel learning [29]). Also, sometimes the connections are just a re-interpretation of existing paradigms [9], which is theoretically interesting but not advantageous in terms of learning better models. Last, but not least, in all works [5, 32, 38, 21, 19, 14, 15] the network adopted are deep, which actually makes the overall pipeline difficult to train, since overfitting must be controlled, the problem of local minima and saddle points must be faced, massive training data is required, as well as expensive hardware resources.

Paper Contributions

Within the previous context, our paper proposes the following main contribution.

1. Within the existing literature of data-driven representation from structured input matrix [5, 32, 38, 21, 19, 14, 15], we propose a novel network architecture, fed by covariance representation which ultimately intertwines hand-crafted kernel methods with data-driven feature learning.

2. We posit that when action's kinematics is properly encoded through a kernel, there is no need to train deep architectures. Shallow networks are indeed effective.

3. We confirm the previous intuition within a broad experimental evaluation over 6 publicly available datasets in 3D action recognition scoring a favorable performance in terms of improvement over state-of-the-art classification methods.

4. We can recover from the scalability issue of kernel methods and mitigate the training issues of neural networks. Therefore, we achieve a strong performance and training/inference efficiency on CPU, ultimately devising an effective action recognition system for the open domain.

Paper outline.

In Section 2, we describe the proposed approach, called *Log-COV-Net*, for 3D action recognition from skeletal data. Section 3 presents a broad experimental evaluation, while Section 4 provides a comprehensive discussion of them. Finally, Section 5 draws conclusions, highlights limitation and profiles future work.

2. *Log-COV-Net*: Log-Covariance Network

Covariance-based representations for action recognition from skeletal joints have attested a superior performance [16, 11, 37, 10, 3]. However, in order to fully exploit the structure which is induced by the covariance representation, classifiers have to be kernelized in order to fully exploit Riemannian geometry when learning decision boundaries to discriminate across different actions. Despite this being mathematically fine, some computational drawbacks arise. Indeed, training such a classifier often requires the computation of Gram matrices, whose quadratic complexity in terms of data instances makes the whole procedure intractable in the big data regime.

On the other side, feature learning approaches via neural networks fully benefit from a gigantic amount of training examples to optimize the huge number (millions, billions) of parameters present in a deep network. At the same time, this is the main reason for the astonishing results scored by data-representations and the source of difficulty in effectively training such networks. Indeed, the optimization problem is non-convex, prone to overfitting, requiring acceleration through parallel GPU computation.

Therefore, despite the strong performance provided by either covariance-based or feature learning paradigms, each of them has its own drawbacks (scalability versus difficult training, respectively). To this end, in this work we aim at intertwining covariance-based and feature approaches in order to combine their pros and get rid of the cons. Namely, our unifying approach will achieve state-of-the-art classification, guaranteeing scalability to the big data regime and allowing easy and fast training/inference on CPU. This is possible by leveraging our intuition that, since exploiting the powerful covariance representation to encode action dynamics, there is no need for the network to be deep. In fact, shallow architectures are just enough in mining discriminative patterns for action classification.

In the rest of this Section, we present the proposed approach called Log-Covariance Network, which is sketched in Fig. 1, and we provide and intuition for it.

Log-Covariance Network. For each action instance $\mathfrak{a}$, acquired in the form of the multi-dimensional time series (1), we compute a covariance matrix $\mathfrak{a}$ according to formula (2). Then, we project $\mathbf{X}_{\mathfrak{a}}$ by a logarithm mapping $\log$. By exploiting the eigendecomposition

$$\mathbf{X}_{\mathfrak{a}} = \mathbf{U} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & \lambda_{3J} \end{bmatrix} \mathbf{U}^\top \tag{3}$$

for $\mathbf{X}_{\mathfrak{a}}$, $\log \mathbf{X}_{\mathfrak{a}}$ is trivial to compute in as follows

$$\log \mathbf{X}_{\mathfrak{a}} = \mathbf{U} \begin{bmatrix} \log \lambda_1 & 0 & \dots & 0 \\ 0 & \log \lambda_2 & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & \log \lambda_{3J} \end{bmatrix} \mathbf{U}^\top, \quad (4)$$

since all λ_i are strictly positive. Formally, this is interpreted as a projection over the tangent space [13, 12, 11], which is locally Euclidean, naturally inducing a vectorization which does not corrupt the geometry. Precisely, we define $\mathbf{v}_{\mathfrak{a}}$ to be the vectorization of all diagonal and lower-diagonal entries[1] of $\log \mathbf{X}_{\mathfrak{a}}$: as similarly done in [16, 37, 3] such intermediate representation is fully able to provide and Euclidean (vectorial) representation which keeps the powerfulness of the Riemannian encoding as SPD matrix [13, 11, 12]. Finally, the vector $\mathbf{v}_{\mathfrak{a}}$ is fed into a fully connected (FC) layer, followed by a sigmoid linearity, which is in turn fed into a classification layer where a hinge loss is exploited. We call our network *Log-COV-Net*.

Implementation details. Despite all matrices $\mathbf{X}_{\mathfrak{a}}$ are positive definite in theory, due to numerical issues, the computed eigenvalues are not always positive: before applying the log mapping, we replace λ_i with $\lambda_i' = \lambda_i + 10^{-4}$. With respect to Fig. 1, note that the "covariance" and "logarithm" layers (which implement equation (2) and (4), respectively) are parameter-free. The only parameter to be trained are the weights $\mathbf{W}$ of the fully connected layer and, of course, the ones of the final classification layer. In our experimental setup, we found that if we jointly train $\mathbf{W}$ and the classifier's parameters, we are highly sensitive to the size of the FC layer. Differently, we achieve more stability by pretraining the FC weights with a cross-entropy loss, also exploiting the powerfulness of supervision. For doing that, we use conjugate gradient descent for all experiments except the ones on NTU-RGB+D [32] dataset (Section 3.6) where we exploit ADAM optimizer with mini-batches of 1024 elements. As a final step, we separately train the hinge-loss classification layer (using libLINEAR [7]).

3. Experiments

We present here the classification accuracies registered by the proposed *Log-COV-Net* in a broad comparison with several state-of-the-art methods on a plethora of benchmark datasets. Namely, we evaluated on MSR-Action3D [20], MSR-Action-Pairs (MSR-*pairs*) [27], Gaming-3D (G3D) [2], Florence3D [31], UTKinect [40], MSRC-Kinect12 [8], HDM-05 [25] and NTU RGB+D [32]. In all cases, we followed the respective recommended training/testing protocols.

[1]Due to simmetry, the upper-diagonal elements are the same as the lower-diagonal ones

As a common preprocessing steps, we compute the relative difference of each joint' triplets $[x_i(t), y_i(t), z_i(t)]^\top$ with the position of the root joint $[x_{\text{root}}(t), y_{\text{root}}(t), z_{\text{root}}(t)]^\top$ for any t. Typically the hip center is adopted as the root. This reduces the actual dimension of $\mathbf{X}_{\mathfrak{a}}$ to $3(J-1) \times 3(J-1)$. Also, the size of the FC layer was cross-validated through grid search within 8, 16, 36, 64, 128, 256 and 512.

3.1. MSR-Action 3D

The dataset consists of 20 actions from 10 different subjects, and is collected with a depth sensor. Each subject performed every action twice or more (total 557 sequences). The 3D locations of 20 joints are provided with the dataset. This is a challenging dataset because many of the actions are highly similar to each other.

Comparative analysis We benchmarked the proposed (*Log-COV-Net*) against the Hankel-based approaches [22, 43], used in tandem with either a Hidden Markov Model (HMM) or a Riemannian-nearest neighbors classifier with prototypes (Hankel-NN-proto). Also, we compared against the tensor representation provided by [18] in using Sequence and Dynamics Compatibility Kernels (SCK+DCK). For covariance based approaches, we also included Kernelized-COV, the kernelization proposed by [3]. We considered the kernel networks of [37] where trial-specific Gram matrix are fed in a second-level kernel responsible for ultimate classification. Finally, we included the LSTM-based approach of [21] where a graph represents skeleton's geometry.

Our experimental findings on MSR-Action-3D are reported in Table 1.

Hankel-HMM [22]	89.0%
SCK + DCK [18]	94.0%
Hankel-NN-proto [43]	94.7%
graph-joint-LSTM [21]	94.8%
Kernelized-COV [3]	96.8%
Ker-RP-RBF [37]	96.9%
Log-COV-Net (proposed)	***97.4%***

Table 1. Evaluation on MSR-Action-3D using the protocol of [20].

3.2. Gaming 3D

The dataset includes 20 different gaming actions like golf swing, tennis serve or bowling. 10 subjects were involved in the acquisition, each of them performing each action three or more times for a total of 663 action sequences, represented by the displacement in time of 20 joints.

Comparative analysis We benchmarked with [26] which combined Restricted Boltzmann machines and Hidden Markov Models. Also, we included several Lie geometry-based methods to encode roto-translations: the shallow ap-

proaches Lie Group [35] and Lie Algebra [36] and the intertwined Lie/deep method [15]. For the latter, we selected LieNet-3Block which is the best performing architecture within the ones proposed in [15].

Results are reported in Table 2.

RBM + HMM [26]	86.4%
LieNet-3Blocks [15]	89.1%
Lie Algebra [36]	90.9%
Lie Group [35]	91.1%
Log-COV-Net (proposed)	***93.0%***

Table 2. Evaluation on Gaming 3D using the protocol of [36].

3.3. UT Kinect

This dataset was captured using a stationary Kinect sensor, the 3D locations of 20 joints are provided. 10 different subjects perform 10 different actions (twice each). This is a challenging dataset due to variations in the view point and high intra-class variations.

Comparative analysis Our proposed *Log-COV-Net* is compared against Lie Group representation [35], HMM fed with Hankel matrices [22], PCA on manifold [1] and the LSTM with graph-based encoding of human skeleton [21]. In addition, we also compared with the reformulation of Histogramd of Oriented Gradients (HOG) features for joints [41] and the aggregation of local spatio-temporal features extracted from raw data [44].

Results are reported in Table 3.

Histograms of 3D joints [41]	90.9%
Spatio-temporal local features [44]	87.9%
Lie Group [35]	97.1%
Hankel-HMM [22]	86.8%
Manifold PCA [1]	94.9%
graph-joint-LSTM [21]	97.0%
Log-COV-Net (proposed)	***98.3%***

Table 3. Evaluation on UT Kinect using the protocol of [35].

3.4. MSRC-Kinect 12

An acquisition of six hours and 40 minutes involves 30 people performing 12 gestures. In total, 6,244 gesture instances. The motion files contain Kinect estimated trajectories of 20 joints.

Comparative analysis Several covariance-based approaches are compared against the proposed *Log-COV-Net*. Precisely, we considered the Bregman divergences for the infinite dimensional operators [11], the temporal pyramid of covariance descriptors [16] and the kernelization recently provided by [3]. Also, we included Ker-RP-POL and Ker-RP-RBF, the two kernel networks of [37].

Results are reported in Table 4.

Bregman-div [11]	89.9%
Ker-RP-POL [37]	90.5%
Ker-RP-RBF [37]	92.3%
Pyramid of COV [16]	93.6%
Ker-COV [3]	95.0%
Log-COV-Net (proposed)	***98.5%***

Table 4. Evaluation on MSRC Kinect 12 using the protocol of [16].

3.5. HDM-05

This dataset contains more than tree hours of systematically recorded and well-documented motion capture data using a 240Hz VICON system to acquire the gestures of 5 non-professional actors via 31 markers. Motion clips have been manually cut out and annotated into roughly 100 different motion classes: on average, 10-50 realizations per class are available. In order to be consistent with the literature, we both replicate the 14 classes evaluation [37, 3] and report the results on the whole dataset.

	14-classes	*all-classes*
sparse-D-SPD [13]	76.1%	N.A.
COV-discriminative [39]	79.8%	N.A.
SPD-dim-red [12]	81.9%	40.0%
Bregman-divergence [11]	82.5%	N.A.
Hankel-NN-proto [43]	86.3%	N.A.
Region-COV [34]	91.5%	58.9%
Ker-RP-POL [37]	93.6%	64.3%
Ker-RP-RBF [37]	96.8%	66.2%
Log-COV-Net (proposed)	***99.1%***	***72.0%***

Table 5. Evaluation on HDM-05 using the two protocols of [37].

Comparative analysis In a broad experimental validation of *Log-COV-Net*, we considered the sparse coding with dictionary learning for SPD matrices of [13], the covariance discriminative learning framework of [39], and the dimensionality reduction technique of [12] for SPD matrices. In addition to Bregman-divergence of the infinite covariance representation [11] and the fast region covariance descriptor of [34], we included the trial-specific encoding of an action with a Gram matrix [37] reporting both Ker-RP-POL and Ker-RP-RBF (polynomial vs. Gaussian RBF kernel). We also compared against [43].

Resultsare reported in Table 5, adding the performance of our *Log-COV-Net* to the published results of [37, Table 4.].

3.6. NTU RGB+D

This huge dataset contains 60 different action classes including daily, mutual, and health-related actions. 40 subjects where involved in the acquisition, for a total number of about 60K instances, captured from 3 different views.

According the suggested experimental protocols [32], we performed a cross validation by testing the model on either different subjects or view with respect to the ones used in training.

Comparative analysis We compared the proposed Log-COV-Net on the NTU-RGB+D dataset. We benchmarked the approaches [42, 28] which rely on normal vector computations, either with a temporal pooling of 3D normals [28] or with modeling of 3D+time spatio-temporal coordinates as a whole [42]. We included the generalization of HOG for skeletal joints [41], also considering the aggregation of raw joints data by means a Gaussian mixture model and Fisher Vectors extraction. We additionally reported the performance of Lie geometry representation by either directly employing Lie Group structure [35] or the Riemannian-training of a neural network [15].

	cross-subject	*cross-view*
Histogram of 3D Normals [28]	30.6%	7.3%
4D Normal Vectors [42]	31.8%	13.6%
Histograms of 3D joints [41]	32.4%	22.3%
Fisher Vectors [6]	38.6%	41.4%
Lie Group [35]	50.1%	52.8%
HB-RNN [5]	56.3%	64.0%
joint-RNN [32]	59.1%	64.1%
LieNet-3Blocks [15]	61.4%	67.0%
joint-LSTM [32]	60.7%	67.3%
graph-joint-LSTM [21]	**69.2%**	**77.7%**
Log-COV-Net (proposed)	*60.9%*	*63.4%*

Table 6. Evaluation on NTU RGB+D with two protocols of [32].

Actually, the release of NTU-RGB+D in 2016 promoted a big boost in training deep architectures for action recognition from skeletal joints. Within the most effective neural network approaches, Recurrent Neural Network (RNN) play a pivotal role. Indeed, [32] trained a RNN by directly feeding raw skeletal data. [5] performed a hierarchical decomposition of human skeleton into arms-legs-torso, modeling each of them with a network and fusing all scores in a bottom-up fashion. Long-short term memory units boosts RNN: [32] used them from raw joints and [21] directly encoded the skeletal geometry by means of a graph.

Results are reported in Table 6

4. Discussion

In this Section we analyze all the results scored on the datasets we consider (Tables 1, 2, 3, 4, 5 and 6). The discussion is carried on according to the cardinality of the training sets, as provided in Table 7.

$N \sim 10^2$ – In the small data regime, the amount of examples available does not allow to fully benefit from the learning from data paradigm. Nevertheless, our proposed *Log-COV-Net* is performing on par with respect to the best method reported hereby (0.2% negative gap on Florence 3D), while improving all baselines in all other cases with about 1% on average.

$N \sim 10^3$ – Increasing by factor 10, we achieve a medium data regime which attested to be the ideal setting for our proposed shallow network. In such a case we register outstanding improvements of *Log-COV-Net* over the state-of-the-art: +3.5% on MSRC-Kinect 12 and +5.8% in the all-class case for HDM-05. This is a strong empirical evidence that the combination of a powerful temporal encoding (through covariance) allows a shallow net to achieve a top performance.

$N \sim 10^4$ – When moving to the big data regime, we have lots of training data and the relatively small number of free parameters in *Log-COV-Net* does not fully capture all available discriminants. Indeed, *Log-COV-Net* is quite gapped by LSTM architectures[2] [32, 21]. In spite of that, we can nevertheless see that all hand-crafted approaches [42, 28, 41, 6, 35] are greatly outperformed in performance, but also scoring on par with respect to alternative deep architectures (e.g. [15] on *cross-subject* protocol or the hierarchical RNN on the *cross-view*). Again, an effective kinematic encoding allows a shallow net to score a strong overall performance.

One further reason for our method to be appealing for open domain action recognition systems is the computational efficiency. Indeed, we adopt a very different perspective from main approaches in the literature. Indeed, we avoid dictionary-based or general pooling aggregation techniques (which slow training) or expensive computational pre-processing such as temporal warping of sequences in order to achieve a fixed temporal length [35, 36]. Additionally, we can simply run our training/inference stage on CPU: 20-30 minutes for training *Log-COV-Net* on NTU RGB+D [32], with almost realtime inference. If compared to [32, 21, 15, 5], the training time in this case is much longer even if using GPU acceleration. Last, but not least, we achieve a quite compact feature representation (the size of the FC is 512 at maximum), which is much much smaller with respect to other approaches, such as [18] or [35].

Thanks to such a compact representation, paired with an extreme training/computational efficiency, we bring strong evidence of the effectiveness of the proposed Log-COV-Net.

5. Conclusion, Limitations and Future Work

In this work we intertwine kernel methods and feature learning by proposing *Log-COV-Net*, a shallow network

[2]Note that, on the small data regime, our *Log-COV-Net* is superior to this architectures: *e.g.*, Tab. 3, +1.3% on graph-joint-LSTM [21].

MSR-Action 3D	Gaming 3D	UT Kinect	HDM-05, *14-classes*
$N \sim 10^2$	$N \sim 10^2$	$N \sim 10^2$	$N \sim 10^2$
$\iota =$ **+0.5%**	$\iota =$ **+1.9%**	$\iota =$ **+1.3%**	$\iota =$ **+2.3%**
MSRC-Kinect12	HDM-05, *all-classes*	NTU-RGB+D, *cross-subject*	NTU-RGB+D, *cross-view*
$N \sim 10^3$	$N \sim 10^3$	$N \sim 10^4$	$N \sim 10^4$
$\iota =$ **+3.5%**	$\iota =$ **+5.8%**	$\iota =$ -8.3%	$\iota =$ -14.3%

Table 7. Comprehensive evaluation of the proposed approach, measuring the (positive or negative) improvement ι of the proposed *Log-COV-Net* with respect to the best among the reported methods. For any dataset, we also provide N, that is the order of magnitude of the available instances.

fed with log-projected covariance representation of skeletal joints data for action recognition.

We empirically prove that, after a powerful structured encoding of action dynamics, there is no actual need to train deep networks for achieving state-of-the-art performance, being a shallow configuration simply enough. Such finding results in an extremely optimized pipeline which can be trained on CPU very fast, performing action classification very efficiently and also relying on a much more compact data representation.

Despite the overall performance is good on the small data regime (hundreds of examples) and remarkable on thousands of instances, one additional order of magnitude (10^4) makes our *Log-COV-Net* suffer with respect to the more elaborated LSTM (which are yet more difficult to train than our network).

Therefore, as a future work, we intend to fill this gap, still preserving compactness of representation and efficiency for training/inference on CPU.

References

[1] R. Anirudh, P. Turaga, J. Su, , and A. Srivastava. Elastic functional coding of human actions: From vector-fields to latent variables. In *CVPR*, 2015. 2, 5

[2] V. Bloom, D. Makris, and V. Argyriou. G3D: A gaming action dataset and real time action recognition evaluation framework. In *CVPR*, 2012. 4

[3] J. Cavazza, A. Zunino, M. San Biagio, and V. Murino. Kernelized covariance for action recognition. In *ICPR*, 2016. 2, 3, 4, 5

[4] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, and A. D. Bimbo. 3d human action recognition by shape analysis of motion trajectories on riemannian manifold. In *IEEE Transactions on Cybernetics*, 2014. 2

[5] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015. 2, 3, 6

[6] G. Evangelidis, G. Singh, and R. Horaud. Skeletal quads: Human action recognition using joint quadruples. In *ICPR*, 2014. 2, 6

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874, 2008. 4

[8] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin. Instructing people for training gestural interactive systems. In *ACM-CHI*, 2012. 4

[9] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*, 2015. 2, 3

[10] M. Ha Quang, M. San Biagio, L. Bazzani, and V. Murino. Approximate log-Hilbert-Schmidt distances between covariance operators for image classification. In *CVPR*, 2016. 2, 3

[11] M. Harandi, M. Salzmann, and F. Porikli. Bregman divergences for infinite dimensional covariance matrices. In *CVPR*, 2014. 2, 3, 4, 5

[12] M. T. Harandi, M. Salzmann, and R. Hartley. From manifold to manifold: geometry-aware dimensionality reduction for SPD matrices. In *ECCV*, 2014. 4, 5

[13] 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 *ECCV*, 2012. 4, 5

[14] Z. Huang and L. V. Gool. A riemannian network for spd matrix learning. In *AAAI*, 2017. 3

[15] Z. Huang, C. Wan, T. Probst, and L. V. Gool. Deep learning on lie groups for skeleton-based action recognition. In *arXiv:1612.05877*, 2016. 2, 3, 5, 6

[16] M. Hussein, M. Torki, M. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. *IJCAI*, 2013. 2, 3, 4, 5

[17] G. Johansson. Visual perception of biological motion and a model for its analysis. 1973. 1

[18] P. Koniusz, A. Cherian, and F. Porikli. Tensor representation via kernel linearization for action recognition from 3d skeletons. In *ECCV*, 2016. 4, 6

[19] C. Li, Y. Hou, P. Wang, and W. Li. Joint distance maps based action recognition with convolutional neural network. *IEEE Signal Process. Lett.*, 2017. 3

[20] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *CVPR workshop*, 2010. 4

[21] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal LSTM with trust gates for 3D human action recognition. In *ECCV*, 2016. 2, 3, 4, 5, 6

[22] L. Lo Presti, M. La Cascia, S. Sclaroff, and O. Camps. Gesture modeling by Hanklet-based hidden Markov model. In *ACCV*, 2014. 2, 4, 5

[23] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks. In *NIPS*, 2014. 2, 3

[24] P. Morerio, J. Cavazza, R. Volpi, R. Vidal, and V. Murino. Curriculum dropout. In *arXiv:1703.06229*, 2017. 2

[25] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. HDM-05 doc. In *Tech. Rep.*, 2007. 4

[26] S. Nie and Q. Ji. Capturing global and local dynamics for human action recognition. In *ICPR*, 2014. 4, 5

[27] O. Oreifej and Z. Liu. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *CVPR*, 2013. 4

[28] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented normals for activity recognition from depth sequences. In *CVPR*, 2013. 6

[29] I. Rebai, Y. BenAyed, and W. Mahdi. Deep multilayer multiple kernel learning. *NCA*, 27(8), 2016. 2, 3

[30] L. Seidenari, V. Varano, S. Berretti, A. D. Bimbo, and P. Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *CVPRw*, 2013. 2

[31] L. Seidenari, V. Varano, S. Berretti, A. D. Bimbo, and P. Pala. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *CVPR workshops*, 2013. 4

[32] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016. 2, 3, 4, 6

[33] Y. Tang. Deep learning using support vector machines. In *ICML workshop*, 2013. 2, 3

[34] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, 2006. 5

[35] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, June 2014. 2, 5, 6

[36] R. Vemulapalli and R. Chellapa. Rolling rotations for recognizing human actions from 3d skeletal data. In *CVPR*, 2016. 2, 5, 6

[37] L. Wang, J. Zhang, L. Zhou, C. Tang, and W. Li. Beyond covariance: Feature representation with nonlinear kernel matrices. In *ICCV*, 2015. 2, 3, 4, 5

[38] P. Wang, Z. Li, Y. Hou, and W. Li. Action recognition based on joint trajectory maps using convolutional neural networks. In *ACMMM*, 2016. 3

[39] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to imageset classification. In *CVPR*, 2013. 5

[40] L. Xia, C.-C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *CVPR workshops*, 2012. 4

[41] L. Xia, C. C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPRw*, 2012. 2, 5, 6

[42] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. In *CVPR*, 2014. 2, 6

[43] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *CVPR*, 2016. 2, 4, 5

[44] Y. Zhu, W. Chen, and G. Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *CVPRw*, 2013. 5