Nonlinear Cross-View Sample Enrichment for Action Recognition

Ling Wang, Hichem Sahbi

Institut Mines-Télécom; Télécom ParisTech; CNRS LTCI

Abstract. Advanced action recognition methods are prone to limited generalization performances when trained on insufficient amount of data. This limitation results from the high expense to label training samples and their insufficiency to capture enough variability due to viewpoint changes.
In this paper, we propose a solution that enriches training data by transferring their features across views. The proposed method is motivated by the fact that cross-view features of the same actions are highly correlated. First, we use kernel-based canonical correlation analysis (CCA) to learn nonlinear feature mappings that take multi-view data from their original feature spaces into a common latent space. Then, we transfer training samples from source to target views by back-projecting their CCA features from latent to view-dependent spaces.
We experiment this cross-view sample enrichment process for action classification and we study the impact of several factors including kernel choices as well as the dimensionality of the latent spaces.

Keywords: Action recognition, kernel methods, canonical correlation analysis, viewpoint knowledge transfer, sample enrichment.

1 Introduction

Human action recognition is highly important for video understanding. In a wide range of applications (such as robotic vision, autonomous driving, video surveillance and retrieval), automatic solutions are necessary in order to recognize several categories of human actions. To achieve this goal, machine learning and classification methods are used to obtain models from existing labeled video collections. However, with the fast growth of industrial applications, action recognition solutions should handle realistic scenarios in challenging conditions including outdoor environment, moving platforms, cluttered background and viewpoint change. As learning and classification methods usually require labeled data, which are scarce and expensive to collect, making use of training data adequately becomes essential.

Under these constraints, enhancing the generalization ability of learning models is necessary, even when labeled training data are scarce. In current literature, action recognition solutions are categorized according to their data representation and learning methods. These works include but not limited to designing discriminative and robust features [7, 19, 31], building compact and effective

representations [37, 28], modeling context and hierarchies [14, 22] and designing kernels [4, 5, 33]. However, even advanced action recognition models have limited generalization power if training data are scarce; indeed, insufficient training samples do not capture enough the inherent variability due to several factors including viewpoint changes.

One possible solution to address this issue is to increase the size of training data by providing larger datasets that sufficiently cover the variability in action recognition; for instance [17, 16, 30, 10] provided around 100 video clips for each action. As training needs to process more videos, this opens a direction to large scale video processing, where models should sufficiently cover the variability. Alternative solutions, based on transfer learning and domain adaption [26], rely on another principle; knowledge learned in one task is applied to another different task in order to make best use of current data. In this framework, no new data needs to be collected, knowledge, however, is added.

In this work, we are interested in solutions that enrich training video data by transferring knowledge about their acquisition conditions (mainly viewpoints). Inspired from the observation that cross-view video features are highly and non linearly correlated, we use a kernel version of canonical correlation analysis (CCA), in order to learn cross-view transfer mappings that take video features from existing (source) to new (target) views.

1.1 Related Work

As action recognition is usually based on appearance and motion features, it is well understood in the literature that large viewpoint change usually causes large variation in these features and reduces the generalization ability of recognition models. The issue of viewpoint change has received a particular attention in action recognition research (see for instance [11, 1]) and several existing techniques [35, 15, 20, 39, 13] focus on building view invariant representations while others combine models learned for different source views [3, 34]. In studies of features through different viewpoints, appearance and motion patterns are shown to be very correlated. In [23], view-dependent vocabularies are connected through corresponding action pairs to build a new dictionary which is more tolerant to view changes. Recently, existing methods make effort to transfer knowledge between viewpoints; several works [21, 38, 36] explore linear relationship between features in fixed source view - target view pairs. Combination models are also needed for view-independent classification. Still, these methods require large quantity of stereo vision data and adequate labeling and annotation information through different viewpoints which are very expensive. Other techniques transfer knowledge through different types of databases. For example, [8] builds a multi-view spatio-temporal AND-OR graph model from 3D human skeleton data and [32] aligns video trajectories with the projected trajectories from a large 3D motion capture database and synthesizes multi-view training data.

1.2 Motivation and Contribution

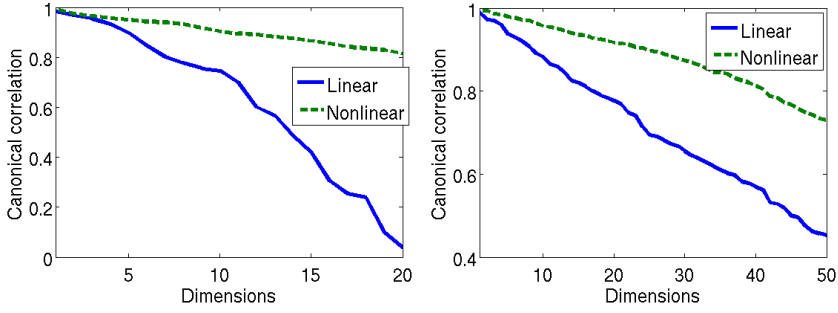
Again, recognizing actions in video is usually based on local appearance and motion features. However, the latter are subject to strong variations due to acquisition conditions (viewpoint in particular). Features, especially motion ones, are not viewpoint invariant, so models learned with insufficient training sets are clearly unable to capture the inherent viewpoint variability and thus have weak generalization power on test data. A straightforward solution to overcome this limitation, is to synthesize large training sets of videos, to better model variability due to viewpoint changes, for instance, by horizontally flipping frames with symmetric views or by using still or animated 3D models. In contrast to our proposed method (see §2), this large scale video synthesis process makes the pre-processing step (prior to train action classifiers) very time demanding as the whole feature extraction pipeline needs to be applied on all the newly generated videos. Furthermore, and regardless processing time issues, this process may hit two majors limitations: i) the insufficiency of the simple flipping operations, ii) and the possible unrealistic aspect of (rendered and animated) 3D models.

Stereo vision is an alternative that provides knowledge in order to learn cross-view mappings. This knowledge is transferable to new videos and may improve classification performance. Usual methods rely on large view-specific stereo datasets to learn transfer models, but during testing, they are not scalable to general datasets whose views (or poses) are not known a priori. Indeed, view-specific data transfer methods may require a preliminary step of human pose estimation in order to decide which view-specific model to apply. Besides the issue of learning pose-specific models[1], pose estimation should also be achieved during the transfer process (i.e., during testing). In contrast, our data transfer solution, presented in this paper, neither requires pose estimation nor view-specific training models for action recognition.

Research on cross-view action recognition proves that multiple-view shots are highly correlated both in appearance and in motion; an assumption of *linear* relationship is helpful (even though insufficient) to learn cross-view transfer models [21, 38, 36]. However, when measuring canonical correlations between cross-view features using CCA (see a particular example in Fig. 1), we observe that they decrease fast, and only few pairs of canonical basis vectors can be found so that projected data are well correlated. With *nonlinear* (kernel-based) CCA, correlations have higher values (see again Fig. 1), and this suggests that relationships between cross-view features are nonlinear.

Considering the above motivations, we introduce in this paper a novel method that enriches video data by transferring their features from few existing training videos (taken from source views) to other views. The proposed method is based on a nonlinear version of canonical correlation analysis and it is motivated by the fact that actions observed across views are highly and non linearly correlated. Using this principle, we show that this feature transfer and enrichment process is highly effective in order to improve the performance of action recognition. This

[1] that may require a lot of human effort in order to label videos and their poses (which is also subject to error).



(a) Velocity feature (b) HOF feature

Fig. 1: A comparison between linear and nonlinear canonical correlation analysis: for linear CCA, canonical correlations decrease fast and only few pairs of canonical basis vectors have highly correlated data while for nonlinear CCA, canonical correlations are high and decrease slowly.

allows us to obtain more training examples that cover many viewpoints especially when the initial training data are insufficient in order to capture the variability due to viewpoint changes. Experiments conducted show the effectiveness of our method for action recognition.

The remainder of this paper is organized as follows: in §2, we propose our cross-view feature transfer using kernel-based CCA; in §3, we present some experiments on action recognition and we compare different kernels and the impact of feature transfer on action recognition. Finally, we conclude our work in § 4.

2 Cross-view Feature Transfer

Consider two overlapping and synchronous video sequences, of the same scenes, taken from two camera viewpoints (referred to as source and target views). We assume that the target view (denoted by t) forms a relative angle θ with respect to the source view (denoted by s). Let $\mathcal{V}_s = \{x_1^s, \ldots, x_n^s\}$, $\mathcal{V}_t = \{x_1^t, \ldots, x_n^t\}$ be the set of features (see §3.1) extracted inside the bounding boxes surrounding moving objects of interest in the source and target views respectively; we assume that $\mathcal{V}_s$, $\mathcal{V}_t$ are ordered so $x_i^s \in \mathcal{V}_s$ is aligned with $x_i^t \in \mathcal{V}_t$, i.e., the underlying features belong to the same moving physical object. In what follows, we use canonical correlation analysis (CCA) in order to learn transformations that *maximize the expected correlation of aligned data in* $\mathcal{V}_s$, $\mathcal{V}_t$ *into a common latent space*. We use these learned transformations to induce (*unobserved*) features in a target view, from *observed* video features taken from a source view.

2.1 Training with Canonical Correlation Analysis

Let $\mathcal{X}$ be an input space (for instance the 96 dimensional HOG space) and consider $\mathcal{V}_s$, $\mathcal{V}_t \subseteq \mathcal{X}$ as two training sets of aligned features. The goal is to learn

transformation matrices $\mathbf{P}_s$, $\mathbf{P}_t$ that characterize features in $\mathcal{V}_s$, $\mathcal{V}_t$ while being viewpoint invariant. Canonical correlation analysis finds two sets of orthogonal axes in $\mathcal{X}$ (also referred to as canonical basis vectors) such that the projection of $\mathcal{V}_s$, $\mathcal{V}_t$, on these axes, maximizes their correlation. Again, $\mathbf{P}_s$, $\mathbf{P}_t$ denote the projection matrices of these orthogonal axes which respectively correspond to views s and t. CCA finds these matrices by maximizing the following criterion [9, 12]:

$$\begin{aligned} (\mathbf{P}_s, \mathbf{P}_t) &= \arg\max_{\mathbf{A},\mathbf{B}} \ \mathbf{A}' C_{st} \mathbf{B} \\ \text{s.t. } \ \mathbf{A}' C_{ss} \mathbf{A} &= 1 \\ \mathbf{B}' C_{tt} \mathbf{B} &= 1 \end{aligned} \tag{1}$$

here $\mathbf{A}'$ stands for transpose of $\mathbf{A}$ and C_{st} (resp. C_{ss}, C_{tt}) is the interclass (resp. intraclass) covariance matrices of data in $\mathcal{V}_s$, $\mathcal{V}_t$. One can show (see for instance [12, 9]) that (1) is equivalent to solving the following eigenproblem:

$$\begin{aligned} C_{st} C_{tt}^{-1} C_{ts} \mathbf{P}_s &= \lambda^2 C_{ss} \mathbf{P}_s \\ \mathbf{P}_t &= \tfrac{1}{\lambda} \ C_{tt}^{-1} C_{ts} \mathbf{P}_s \end{aligned} \tag{2}$$

Projection matrices $\mathbf{P}_s$, $\mathbf{P}_t$ define a common latent space (denoted by $\mathcal{L} \subset \mathbb{R}^d$) in which the correlation between $(\mathbf{P}_s' \ x_i^s, \mathbf{P}_t' \ x_i^t) \in \mathcal{L} \times \mathcal{L}$ is maximized ($i = 1, \ldots, n$). Note that cross-view transformations might not be only related to linear geometric transformations as they include other nonlinear physical aspects including (illumination changes, etc.), so one should consider a nonlinear version of CCA using kernel mapping (see §2.2). Prior to describe the cross-view mapping method in §2.3, we will review kernel mapping via kernel principal component analysis (KPCA) in §2.2. The latter makes it possible to control dimensionality of data and helps defining new mapping spaces so that CCA transformations become nonlinear.

2.2 Kernel Mapping

Let Φ be an implicit mapping (defined via a kernel function $K(x, z) = \Phi(x)'\Phi(z)$) from the input space $\mathcal{X}$ into a high dimensional feature space $\mathcal{H}$. Assume the training set $\mathcal{V}_s$ is centered in $\mathcal{H}$, i.e., $\sum_{i=1}^n \Phi(x_i^s) = 0$. KPCA finds principal orthogonal projection axes by diagonalizing the covariance matrix $M = (1/n) \sum_{i=1}^n \Phi(x_i^s)\Phi(x_i^s)'$. The principal orthogonal axes, denoted $\{E_k\}_{k=1}^n$, can be found by solving the eigenproblem $ME_k = \lambda_k E_k$, where E_k, λ_k are, respectively, the k^{th} eigenvector and its underlying eigenvalue. It can be shown (see for instance [29]) that the solution of the above eigenproblem lies in the span of the training data, i.e., $\forall k = 1, \ldots, n, \ \exists \alpha_{k1}, \ldots, \alpha_{kn} \in \mathbb{R}$ s.t. $E_k = \sum_{j=1}^n \alpha_{kj} \Phi(x_j^s)$, where $\alpha_k = (\alpha_{k1}, \ldots, \alpha_{kn})$ are found by solving the eigenproblem $K\alpha_k = \lambda_k \alpha_k$. Here K is the Gram matrix on the centered data in $\mathcal{V}_s$ in the feature space $\mathcal{H}$. In case the data are not centered, this matrix is defined as

$$K_{ij} = \left\langle \Phi(x_i^s) - \frac{1}{n} \sum_k \Phi(x_k^s), \Phi(x_j^s) - \frac{1}{n} \sum_k \Phi(x_k^s) \right\rangle, \tag{3}$$

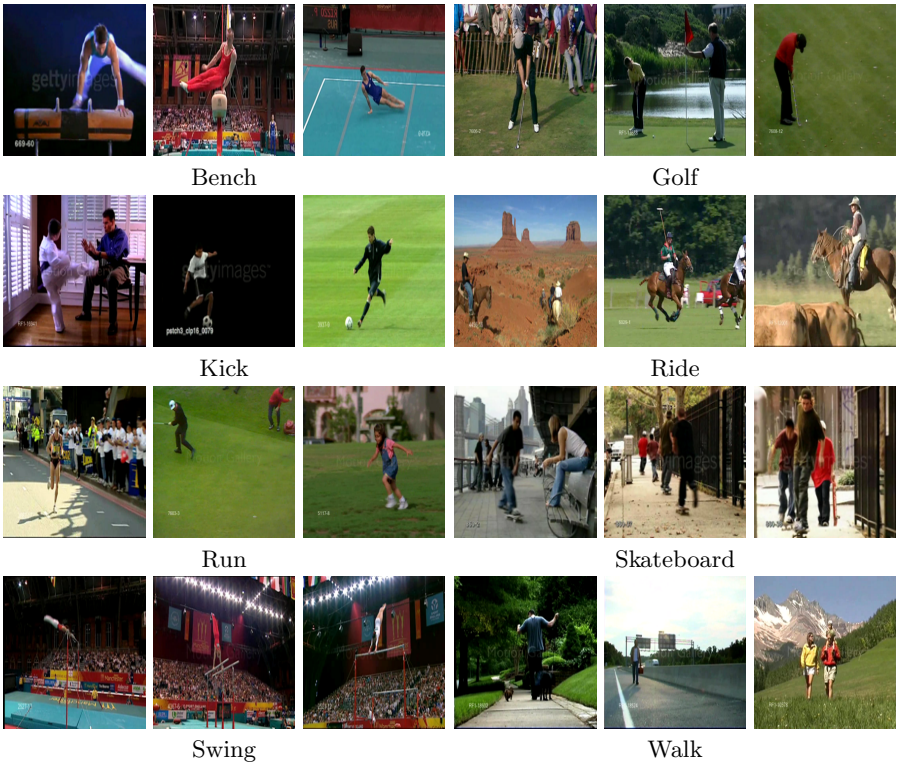


Fig. 2: General action recognition videos in different viewpoints [27].

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Each data $x \in \mathcal{V}_s$ is explicitly mapped into $\psi(x) \in \mathbb{R}^p$, where $\psi(x) = \big(\langle x, E_1 \rangle, \ldots, \langle x, E_p \rangle\big)'$ $(p \ll n)$. The same KPCA mapping process is achieved for data in $\mathcal{V}_t$. CCA is now applied to $\psi(\mathcal{V}_s)$, $\psi(\mathcal{V}_t) \subset \mathbb{R}^p$ as shown subsequently.

2.3 Cross-View Feature Transfer using CCA

Fig. 2 shows video examples of actions taken from different views. A hypothesis of cross-view transfer learning is that features in different views are drawn from the same distribution in the latent space $\mathcal{L}$. So, we assume that latent features capture visual characteristics of moving objects/persons while being tolerant to their viewpoint changes. Thus the features in the input space $\mathcal{X}$ (extracted directly from video data) are connected by the CCA latent features in $\mathcal{L}$. Assuming that mappings $\mathbf{P}_s$, $\mathbf{P}_t$ are invertible (or utilizing Moore-Penrose pseudoinverse [6]), we transfer features $\{\psi(x^s)\}$ (from the source view) to features $\{\psi(x^t)\}$ (in the target view) by

$$\psi(x^t) := (\mathbf{P}_s \mathbf{P}_t^{-1})'(\psi(x^s) - \bar{\psi}^s) + \bar{\psi}^t, \tag{4}$$

here $\bar{\psi}^s$, $\bar{\psi}^t$ are the estimated sample means in $\psi(\mathcal{V}_s)$, $\psi(\mathcal{V}_t)$ respectively. Notice that Eq. (4), follows the assumption that CCA projections in the latent space $\mathcal{L}$ are viewpoint invariant, i.e., $\mathbf{P}'_s(\psi(x^s) - \bar{\psi}^s) \simeq \mathbf{P}'_t(\psi(x^t) - \bar{\psi}^t)$.

In practice, we build transformation matrices $\mathbf{P}_s$, $\mathbf{P}_t$ using a dataset of "source-target" view video sequences that correspond to the same moving actors [17] (see Fig. 3; in this example, (b, c, d, e, f) correspond to the source views while (a, b, c, d, e) correspond to the target view). This dataset includes simple actions shot by 20 cameras with equiangular optical axes; the angle θ between these axes is constant and relatively small ($\theta = 18^0$) in order to avoid occlusions and to obtain enough alignments across views.

We generate abundant trajectories from video data by tracking densely sampled keypoints [31]. As frames in different views are synchronized, we obtain sets $\mathcal{V}_s$, $\mathcal{V}_t$ of aligned trajectories according to their keypoint locations in corresponding frames using SIFT matching [25]. In the learning process, we randomly sample 4000 pairs of corresponding trajectories in $\mathcal{V}_s$, $\mathcal{V}_t$ and use them to learn KPCA mapping as well as CCA transformation matrices $\mathbf{P}_s$, $\mathbf{P}_t$.

2.4 Video Set Enrichment

Using the learned transformation matrices $\mathbf{P}_s$, $\mathbf{P}_t$, we enrich the set of training videos across different views by transferring component features. We describe each video with component-based features using the method in [33] that generates and groups abundant trajectories from video data by tracking densely sampled keypoints. During video enrichment, we first map source view component features (defined as centers of their assigned trajectory features) using KPCA, then we generate new component features associated to a target view, by transferring their KPCA features using Eq. (4). Again, this transfer assumes that corresponding features in the latent space are highly correlated.

Given a set of training videos $\{V_i\}$ with unknown source view angles in $\{\alpha_i\}$, the transfer process "hallucinates" a new set of video component features *with a relative angle* θ (i.e., with target view angles in $\{\alpha_i + \theta\}$); note that generated video features inherit the same labels as original videos. Though we cannot process videos in target views (as they are not available), we can generate new training samples by the transfer process in Eq. 4. In this way, generating many training samples covering more views is very efficient and also effective for action recognition as shown in experiments.

3 Experimental Analysis

In this section, we measure the impact of our cross-view feature transfer method on action recognition. We first evaluate the ability of KPCA[2] in order to generate high dimensional and more discriminating features. Then, we apply CCA on the obtained KPCA features and evaluate its performance for cross-view feature transfer. As will be shown later in this section, the generalization power of

[2] with different kernel functions.

(a) (b) (c) (d) (e) (f)

Fig. 3: Frame examples from stereo videos [24].

our feature transfer method, and its positive impact on action recognition performances, increases when the dimensions of KPCA and CCA mappings reach reasonably large values.

3.1 Evaluation Set and Setting

Dataset and features. We conduct experiments on the UCF Sport dataset [27]. The latter includes 150 videos containing 10 classes of actions shot in real environment with large viewpoint variation and background changes. In order to evaluate the results, we use the data split protocol introduced in [18]; we build support vector machine (SVM) classifiers on the training set (of 103 videos), and report the classification accuracy on the remaining test set (of 47 videos). As a preprocessing step, we first extract dense trajectories [31] from original videos and we keep only those inside the bounding boxes provided by [18], so that trajectories are roughly around the human body[3]. Then, we group these trajectories into mid-level components using the unsupervised clustering method in [33]. Finally, we represent each video by the set of its mid-level components (limited to 100 components in practice) and each one is described by five feature vectors: Velocity Shape, HOG, HOF, MBHx, MBHy (see [31] for details).

Action recognition with SVMs. We use the convolution kernel in order to evaluate the similarity between any two given videos V, V'; this kernel is defined as $K(V,V') = \sum_{c\in V, c'\in V'} k_e(c,c')$. Here two choices are considered for the elementary kernel k_e: linear and Laplacian RBF with $\gamma = 1$ (see table 1). The convolution kernel is computed between pairwise videos and plugged into SVMs for training and classification. In all these experiments, the regularization parameter C in SVMs is fixed to 10 and "one vs. all" SVMs are trained for each action class; a given test video is assigned to the action class that maximizes its SVM score. In the remainder of this section, SVM classifiers based on linear (resp. Laplacian RBF) elementary kernels will be referred to as LCK (resp. RCK).

In what follows, we study the influence of kernel choice used for KPCA on the performance of action recognition. These performances are reported for different

[3] Besides, according to our transfer settings, we only transfer knowledge on actions.

KPCA and CCA dimensions corresponding to the p-largest eigenvalues (of the underlying kernel matrices) and the d-largest correlations (associated to the canonical basis vectors) respectively (see §2.2, §2.1).

	Kernel Type	Formulation	Parameters
1	Linear	$k(x,y) = x'y$	–
2	Polynomial	$k(x,y) = (\gamma x'y + 1)^d$	$\gamma = 1, d = 2$
3	NegDist	$k(x,y) = \|\|x-y\|\|^p$	$p = 1$
4	GHI**	$k(x,y) = \sum_i \min(\|x_i\|^\gamma, \|y_i\|^\gamma)$	$\gamma = 1$
5	Gaussian RBF*	$k(x,y) = e^{-\frac{\|\|x-y\|\|^2}{2\gamma\overline{d}^2}}$	$\gamma \in \{0.01, 1, 100\}$
6	Laplacian RBF*	$k(x,y) = e^{-\frac{\|\|x-y\|\|}{\gamma\overline{d}}}$	$\gamma \in \{0.1, 1, 10\}$

** Generalized Histogram Intersection Kernel [2]

* In RBF like kernels, $\overline{d}$ is the mean distance between all training samples.

Table 1: Types of kernels used in the experiments.

3.2 Influence of KPCA Mapping on Action Recognition

As discussed in §2.3, KPCA is a preprocessing step that maps data from an input space $\mathcal{X}$ into a high dimensional space $\mathcal{H}$ so that cross-view CCA transformations can be learned and applied in $\mathcal{H}$. Note that KPCA mapping is used not only to make CCA transformations nonlinear, but also to make features more discriminating in $\mathcal{H}$. Thus, prior to evaluate the performance of CCA transfer (see §3.3), we evaluate in this section the ability of KPCA to produce more discriminating features in $\mathcal{H}$ by measuring its impact on the performance of action recognition (i.e., without sample enrichment). This impact is measured for different kernel choices (see again table 1) and also for different values of p (the dimension of KPCA mapping).

Table 2 shows action recognition performances for different kernel choices for KPCA (listed in table 1); these results are obtained with linear SVM classifiers (LCK). The results show that nonlinear KPCA mapping improves classification performances (w.r.t linear mapping) especially when using the generalized histogram intersection and RBF kernels. Note that Gaussian RBF kernel is more sensitive to the choice of parameters than Laplacian RBF and the latter has a similar behavior compared to NegDist kernel. These results also corroborate the fact that features in $\mathcal{H}$ are more discriminating when dimension p is sufficiently (but not very) large.

3.3 Influence of CCA Mapping on Action Recognition

Considering aligned training features $\psi(\mathcal{V}_s),\ \psi(\mathcal{V}_t) \subset \mathbb{R}^p$ of the source and target views, we use CCA projection matrices $\mathbf{P}_s$, $\mathbf{P}_t$ (as shown in §2.1, §2.2, §2.3 and

Kernels for KPCA \ KPCA dim (p)	64	128	256	512	1024	2048
Linear (baseline)	53.2	**57.4**	–	–	–	–
Polynomial	59.6	**61.7**	61.7	61.7	61.7	61.7
NegDist	61.7	66.0	**68.1**	68.1	68.1	68.1
GHI	68.1	68.1	**72.3**	72.3	70.2	70.2
Gaussian RBF ($\gamma = 0.01$)	66.0	68.1	**72.3**	70.2	70.2	72.3
Gaussian RBF ($\gamma = 1$)	**59.6**	59.6	59.6	59.6	59.6	59.6
Gaussian RBF ($\gamma = 100$)	**59.6**	59.6	59.6	–	–	–
Laplacian RBF ($\gamma = 0.1$)	66.0	68.1	70.2	70.2	**72.3**	72.3
Laplacian RBF ($\gamma = 1$)	66.0	68.1	68.1	68.1	68.1	**70.2**
Laplacian RBF ($\gamma = 10$)	63.8	66.0	**68.1**	68.1	68.1	68.1

Table 2: This table shows action recognition performance using KPCA mapping and linear SVMs: features are projected into a p dimensional space. Note that we do not explore larger values of p (i.e., $p > 128$) for the linear KPCA as the dimension of the input space is bounded by 128. Similarly, we do not explore larger values of p for the Gaussian RBF KPCA (with $\gamma = 100$) as the latter behaves as linear KPCA for large values of γ.

in Eq. (4)) in order to enrich the training set of videos. The purpose is to show the impact of this CCA-based enrichment process on action recognition.

Dimensionality and kernel choice. Tables 3, 4, 5 illustrate the impact of CCA transfer on action recognition performances for different settings of kernels in KPCA including linear, histogram intersection and Gaussian RBF respectively. These tables also report performances for different values of dimensions p, d (related to KPCA and CCA mapping respectively) and Fig. 4 shows transfer error between generated features and those in the ground truth both in target views, w.r.t p. From these results, it is clear that better performances and small transfer errors are obtained with nonlinear KPCA mappings, particularly with histogram intersection, and these performances increase as the dimensions p, d of KPCA and CCA mappings become reasonably large, both with linear and nonlinear SVM classifiers. Indeed, when d is small, canonical basis vectors (in $\mathbf{P}_s$, $\mathbf{P}_t$) preserve accurate relationship between transferred features (i.e., high correlations) while less knowledge is transferred. As more dimensions are taken (i.e., as d increases), more knowledge is transferred but with more bias due to the decrease of the canonical correlations.

Motion vs. appearance features. In order to understand the importance of transferred (motion[4] and appearance[5]) features, we compare two settings: in the first one i) we transfer only motion features and we consider appearance features

[4] Velocity Shape, HOF, MBHx, MBHy.
[5] HOG.

	Linear SVMs (LCK)			Nonlinear SVMs (RCK)		
	noenrich	enrich perfs w.r.t d		noenrich	enrich perfs w.r.t d	
$p \backslash d$		$d = 64$	$d = 128$		$d = 64$	$d = 128$
$p = 64$	53.2	**59.6**	-	76.6	**80.9**	-
$p = 128$	57.4	55.3	**61.7**	78.7	**80.9**	80.9

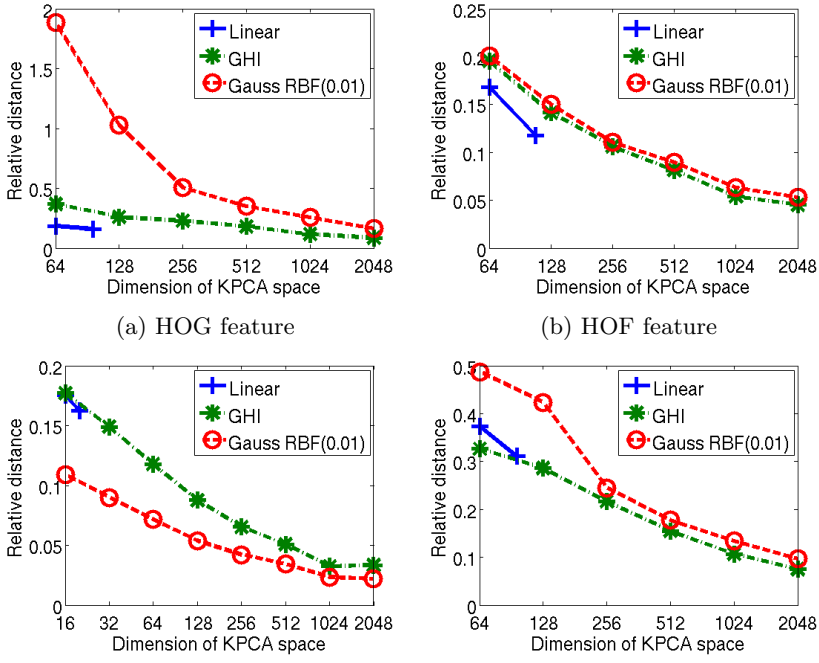
Table 3: This table shows action recognition performances (%) with and without the enrichment process for different values of p (related to linear KPCA mapping) and d (related to CCA).

	Linear SVMs (LCK)							Nonlinear SVMs (RCK)						
	noenrich	enrich perfs w.r.t d						noenrich	enrich perfs w.r.t d					
$p \backslash d$		64	128	256	512	1024	2048		64	128	256	512	1024	2048
64	68.1	59.6	–	–	–	–	–	78.7	**83.0**	–	–	–	–	–
128	68.1	**70.2**	66.0	–	–	–	–	74.5	**80.9**	76.6	–	–	–	–
256	72.3	72.3	**76.6**	68.1	–	–	–	74.5	76.6	**78.7**	78.7	–	–	–
512	72.3	**74.5**	74.5	70.2	68.1	–	–	74.5	70.2	72.3	**78.7**	70.2	–	–
1024	70.2	**74.5**	72.3	70.2	74.5	68.1	–	74.5	**76.6**	74.5	76.6	76.6	70.2	–
2048	70.2	70.2	70.2	**72.3**	72.3	70.2	72.3	74.5	72.3	72.3	74.5	72.3	66.0	66.0

Table 4: This table shows action recognition performances (%) with and without the enrichment process for different values of p (related to histogram intersection KPCA mapping) and d (related to CCA). Note that $d \leq p$ as the dimension of CCA cannot exceed that of KPCA.

	Linear SVMs (LCK)							Nonlinear SVMs (RCK)						
	noenrich	enrich perfs w.r.t d						noenrich	enrich perfs w.r.t d					
$p \backslash d$		64	128	256	512	1024	2048		64	128	256	512	1024	2048
64	66.0	59.6	–	–	–	–	–	70.2	**72.3**	–	–	–	–	–
128	68.1	68.1	63.8	–	–	–	–	72.3	72.3	72.3	–	–	–	–
256	72.3	72.3	70.2	66.0	–	–	–	72.3	**76.6**	74.5	76.6	–	–	–
512	70.2	70.2	**72.3**	70.2	72.3	–	–	72.3	74.5	76.6	**80.9**	72.3	–	–
1024	70.2	**76.6**	76.6	74.5	72.3	70.2	–	72.3	**74.5**	74.5	74.5	72.3	72.3	–
2048	72.3	72.3	74.5	74.5	74.5	**76.6**	72.3	72.3	74.5	**80.9**	68.1	70.2	70.2	70.2

Table 5: This table shows action recognition performances (%) with and without the enrichment process for different values of p (related to Gaussian RBF KPCA mapping, with $\gamma = 0.01$) and d (related to CCA). Note that $d \leq p$ as the dimension of CCA cannot exceed that of KPCA.



(a) HOG feature (b) HOF feature

(c) Velocity feature (d) MBH feature

Fig. 4: This figure shows the trend of transfer error between generated and ground truth features in target views when increasing the dimension p of KPCA mapping; for fixed p, dim d is set to obtain the full rank p. This transfer error is measured using the average relative distance defined as $dist(x, z) := \frac{1}{n}\sum_{i=1}^{n} ||x_i - z_i||/||z_i||$.

	Linear SVMs (LCK)		Nonlinear SVMs (RCK)	
$p \backslash d$	64	128	64	128
64	59.6/**61.7**	-	**80.9**/78.7	-
128	55.3/**61.7**	**61.7**/57.4	**80.9**/78.7	**80.9**/**80.9**

Table 6: This table shows a comparison between "motion and appearance transfer" vs. "motion transfer only" for different values of p, d. In these results linear kernel is used for KPCA. Note that $d \leq p$ as the dimension of CCA cannot exceed that of KPCA.

	Linear SVMs (LCK)						Nonlinear SVMs (RCK)					
$p\backslash d$	64	128	256	512	1024	2048	64	128	256	512	1024	2048
64	59.6/59.6	–	–	–	–	–	**83.0**/80.9	–	–	–	–	–
128	**70.2**/68.1	66.0/63.8	–	–	–	–	**80.9**/**80.9**	76.6/76.6	–	–	–	–
256	72.3/68.1	**76.6**/68.1	68.1/66.0	–	–	–	76.6/78.7	78.7/**80.9**	78.7/78.7	–	–	–
512	**74.5**/68.1	74.5/70.2	70.2/72.3	68.1/66.0	–	–	70.2/76.6	72.3/80.9	78.7/**83.0**	70.2/74.5	–	–
1024	**74.5**/70.2	72.3/70.2	70.2/70.2	74.5/68.1	68.1/70.2	–	76.6/78.7	74.5/**80.9**	76.6/80.9	76.6/78.7	70.2/70.2	–
2048	70.2/**72.3**	70.2/70.2	**72.3**/68.1	72.3/66.0	70.2/68.1	72.3/72.3	72.3/76.6	72.3/74.5	74.5/**78.7**	72.3/70.2	66.0/66.0	66.0/66.0

	Linear SVMs (LCK)						Nonlinear SVMs (RCK)					
$p\backslash d$	64	128	256	512	1024	2048	64	128	256	512	1024	2048
64	59.6/**66.0**	–	–	–	–	–	**72.3**/**72.3**	–	–	–	–	–
128	**68.1**/66.0	63.8/66.0	–	–	–	–	**72.3**/70.2	72.3/**72.3**	–	–	–	–
256	**72.3**/**72.3**	70.2/68.1	66.0/70.2	–	–	–	**76.6**/74.5	74.5/72.3	76.6/74.5	–	–	–
512	70.2/**72.3**	**72.3**/72.3	70.2/70.2	72.3/72.3	–	–	74.5/74.5	76.6/74.5	**80.9**/78.7	72.3/72.3	–	–
1024	**76.6**/74.5	76.6/70.2	74.5/68.1	72.3/72.3	70.2/70.2	–	74.5/74.5	74.5/**76.6**	74.5/72.3	72.3/72.3	72.3/72.3	–
2048	72.3/74.5	74.5/74.5	74.5/76.6	74.5/**76.6**	**76.6**/74.5	72.3/72.3	74.5/76.6	**80.9**/78.7	68.1/68.1	70.2/70.2	70.2/70.2	70.2/70.2

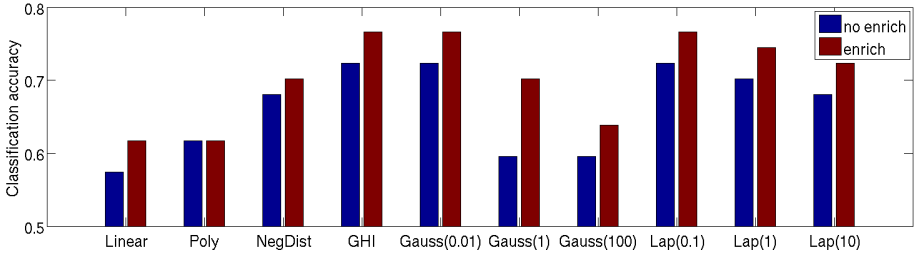
Table 7: This table shows a comparison between "motion and appearance transfer" vs. "motion transfer only" for different values of p, d. In these results histogram intersection kernel (top) and Gaussian RBF kernel, with $\gamma = 0.01$ (bottom) are used for KPCA. Note that $d \leq p$ as the dimension of CCA cannot exceed that of KPCA.

viewpoint invariant, while in the second setting ii) we transfer both motion and appearance features. Tables 6 – 7 compare the impact of these two settings on the accuracy of action recognition; each cell in these tables corresponds to a pair - ("motion and appearance transfer" vs. "motion transfer only"). We observe that the setting (i) (i.e., assuming viewpoint invariant HOG features) is relatively accurate but globally the setting (ii) is more accurate especially when dimensions p, d are not large.

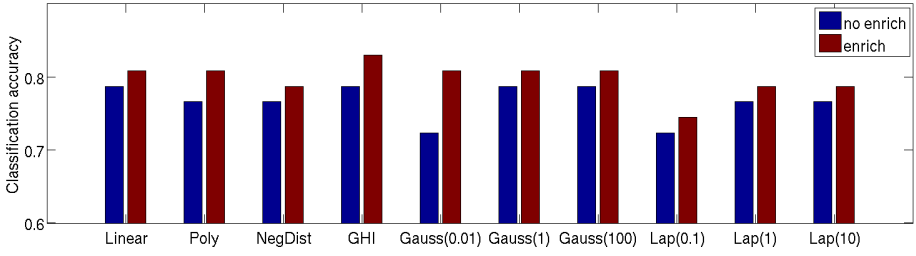
Overall performances. Finally, Fig. 5 compares the overall action classification performance of sample enrichment against no enrichment. For each kernel, we show the best results (corresponding to the best setting of p and d). This figure clearly shows that training SVM classifiers for action recognition, on enriched training set makes performance better than the initial setting that trains classifier without enrichment. Moreover, as shown in Fig. 6, when enough data is used for enrichment, the improvement becomes noticeable.

4 Conclusions

In this paper, we proposed a method to enrich training samples by transferring their features from source to target views. Inspired from the observation that cross-view features are highly and non linearly correlated, we used kernel-based canonical correlation analysis in order to map features across views. Experiments conducted show the positive impact of this enrichment process on action recognition and the influence of different (mainly nonlinear) kernels on the performances.

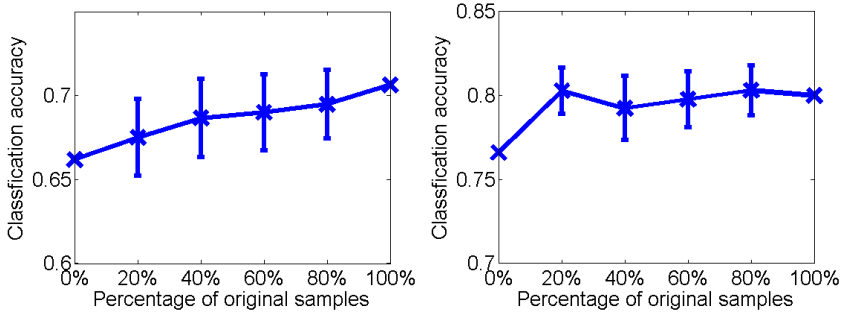


(a) Linear SVMs (LCK)



(b) Nonlinear SVMs (RCK)

Fig. 5: This figure shows action recognition performances with and without the enrichment process for different kernels.



(a) Linear SVMs (LCK) (b) Nonlinear SVMs (RCK)

Fig. 6: This figure shows the evolution of action recognition performances w.r.t the fraction k of original training data involved in enrichment ($k = 0\%$ stands for no enrichment while $k = 100\%$ means that all original data are used for enrichment, thereby the size of training set doubles). These results correspond to the average classification accuracy of 100 runs. Each run corresponds to a fraction k of random training samples used for enrichment.

Acknowledgements

This work was supported in part by a grant from the Research Agency ANR (Agence Nationale de la Recherche) under the MLVIS project and a grant from DIGITEO under the RELIR and VISUNET projects.

References

1. Ashraf, N., Shen, Y., Cao, X., Foroosh, H.: View invariant action recognition using weighted fundamental ratios. Computer Vision and Image Understanding 117(6), 587–602 (2013)
2. Boughorbel, S., Tarel, J.P., Boujemaa, N.: Generalized histogram intersection kernel for image recognition. In: ICIP (2005)
3. Farhadi, A., Tabrizi, M.K.: Learning to Recognize Activities from the Wrong View Point. In: ECCV (2008)
4. Gaidon, A., Harchaoui, Z., Schmid, C.: A time series kernel for action recognition. In: BMVC (2011)
5. Gaidon, A., Harchaoui, Z., Schmid, C.: Activity representation with motion hierarchies. Int. J. Comput. Vision 107(3), 219–238 (2013)
6. Golub, G.H., Van Loan, C.F.: Matrix Computations (3rd Ed.). Johns Hopkins University Press, Baltimore, MD, USA (1996)
7. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as Space-Time Shapes. IEEE Trans. Pattern Anal. Mach. Intell. 29(12), 2247–2253 (2007)
8. Gupta, A., Martinez, J., Little, J.J., Woodham, R.J.: 3D Pose from Motion for Cross-view Action Recognition via Non-linear Circulant Temporal Encoding. In: CVPR (2014)
9. Hardoon, D.R., Szedmak, S.R., Shawe-taylor, J.R.: Canonical Correlation Analysis: An Overview with Application to Learning Methods. Neural Comput. 16(12), 2639–2664 (2004)
10. Hassner, T.: A Critical Review of Action Recognition Benchmarks. In: 1st IEEE International Workshop on Action Similarity in Unconstrained Videos (ACTS) at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2013)
11. Holte, M.B., Tran, C., Trivedi, M.M., Moeslund, T.B.: Human Action Recognition Using Multiple Views: A Comparative Perspective on Recent Developments. In: Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding (2011)
12. Hotelling, H.: Relations Between Two Sets of Variates. Biometrika 28(3/4), 321–377 (1936)
13. Huang, C.H., Yeh, Y.R., Wang, Y.C.F.: Recognizing Actions across Cameras by Exploring the Correlated Subspace. In: ECCV Workshops (1) (2012)
14. Jiang, Z., Lin, Z., Davis, L.: Recognizing Human Actions by Learning and Matching Shape-Motion Prototype Trees. IEEE Trans. Pattern Anal. Mach. Intell. 34(3), 533–547 (2012)
15. Junejo, I., Dexter, E., Laptev, I., Prez, P.: Cross-view action recognition from temporal self-similarities. In: ECCV (2008)
16. Kliper-Gross, O., Hassner, T., Wolf, L.: The Action Similarity Labeling Challenge. IEEE Trans. Pattern Anal. Mach. Intell. 34(3), 615–621 (2012)
17. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A Large Video Database for Human Motion Recognition. In: ICCV (2011)

18. Lan, T., Wang, Y., Mori, G.: Discriminative Figure-Centric Models for Joint Action Localization and Recognition. In: ICCV (2011)
19. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning Realistic Human Actions from Movies. In: CVPR (2008)
20. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR (2011)
21. Li, R., Zickler, T.: Discriminative virtual views for cross-view action recognition. In: CVPR (2012)
22. Liang, X., Lin, L., Cao, L.: Learning Latent Spatio-temporal Compositional Model for Human Action Recognition. In: Proceedings of the 21st ACM International Conference on Multimedia (2013)
23. Liu, J., Shah, M., Kuipers, B., Savarese, S.: Cross-view action recognition via view knowledge transfer. In: CVPR (2011)
24. Liu, Y., Dai, Q., Xu, W.: A Point-Cloud-Based Multiview Stereo Algorithm for Free-Viewpoint Video. IEEE Trans. Vis. Comput. Graph. 16(3), 407–418 (2010)
25. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
26. Pan, Sinno Jialin and Yang, Qiang: A Survey on Transfer Learning. IEEE Trans. on Knowl. and Data Eng. 22(10), 1345–1359 (2010)
27. Rodriguez, M.D., Ahmed, J., Shah, M.: Action MACH: A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition. In: CVPR (2008)
28. Sadanand, S., Corso, J.J.: Action Bank: A High-Level Representation of Activity in Video. In: CVPR (2012)
29. Schlkopf, B., Smola, A.J., Müller, K.R.: Kernel principal component analysis. Advances in kernel methods: support vector learning pp. 327–352 (1999)
30. Soomro, k., Roshan Zamir, A., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. In: CRCV-TR-12-01 (2012)
31. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense Trajectories and Motion Boundary Descriptors for Action Recognition. Int. J. Comput. Vision 103(1), 60–79 (2013)
32. Wang, J., Nie, X., Xia, Y., Wu, Y., Zhu, S.C.: Cross-view Action Modeling, Learning and Recognition. In: CVPR (2014)
33. Wang, L., Sahbi, H.: Directed Acyclic Graph Kernels for Action Recognition. In: ICCV (2013)
34. Weinland, D., Özuysal, M., Fua, P.: Making Action Recognition Robust to Occlusions and Viewpoint Changes. In: ECCV (2010)
35. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Computer Vision and Image Understanding 104(2), 249–257 (2006)
36. Wu, X., Wang, H., Liu, C., Jia, Y.: Cross-view Action Recognition over Heterogeneous Feature Spaces. In: ICCV (2013)
37. Yang, Y., Saleemi, I., Shah, M.: Discovering Motion Primitives for Unsupervised Grouping and One-Shot Learning of Human Actions, Gestures, and Expressions. IEEE Trans. Pattern Anal. Mach. Intell. 35(7), 1635–1648 (2013)
38. Zhang, Z., Wang, C., Xiao, B., Zhou, W., Liu, S., Shi, C.: Cross-View Action Recognition via a Continuous Virtual Path. In: CVPR (2013)
39. Zheng, J., Jiang, Z., Phillips, P.J., Chellappa, R.: Cross-View Action Recognition via a Transferable Dictionary Pair. In: BMVC (2012)