Action Recognition using Canonical Correlation Kernels

G Nagendar, Sai Ganesh, Mahesh Goud and C.V Jawahar

Centre for Visual Information Technology,
IIIT Hyderabad, India

Abstract. In this paper, we propose the canonical correlation kernel (CCK), that seamlessly integrates the advantages of lower dimensional representation of videos with a discriminative classifier like SVM. In the process of defining the kernel, we learn a low-dimensional (linear as well as nonlinear) representation of the video data, which is originally represented as a tensor. We densely compute features at single (or two) frame level, and avoid any explicit tracking. Tensor representation provides the holistic view of the video data, which is the starting point of computing the CCK. Our kernel is defined in terms of the principal angles between the lower dimensional representations of the tensor, and captures the similarity of two videos in an efficient manner. We test our approach on four public data sets and demonstrate consistent superior results over the state of the art methods, including those that use canonical correlations.

1 Introduction

Recent advances in action recognition are propelled by (i) the use of local as well as global features [14, 28], which have significantly helped in object and scene recognition, by computing them over 2D frames [8, 28] or over a 3D video volume [6] (ii) the use of factorization techniques over video volume tensors [9, 15] and defining similarity measures over the resulting lower dimensional factors [2]. In this paper, we try to take advantages of both these approaches by defining a canonical correlation kernel that is computed from tensor representation of the videos. This also enables seamless feature fusion by combining multiple feature kernels.

Wang *et al.* [28] demonstrated the successful use of multiple features defined over relatively densely extracted tracks. Motivated by the success of dense features for object recognition [23, 29], we do a further dense feature extraction on a regular grid of pixels which helps us in obtaining a rich and robust set of descriptors. However, we avoid any explicit tracking across frames. Though there have been many previous attempts in using spatio-temporal descriptors in the past [25], our focus is to explore the utility of well understood 2D image descriptors. Our method, in fact, captures the temporal information as well as correlation across the frames while computing the low-dimensional representations of tensors.

Spatio-temporal shape and texture of the action videos are well represented in a number of low-rank representations computed out of the tensors with various factorization techniques [15, 16]. This tensor computation has been successfully applied in many vision tasks including action classification in the past [9, 15, 16]. Recognition is often carried out on the prominent components obtained by the factorization, or dimensionality reduction of the videos. Kim *et al.* [9] represent the video as a tensor and compute the similarities using canonical correlation [2] with a specially selected discriminative correlation coefficients. These tensorial representation methods [9, 15, 16] often use pixel values directly as observations to build the data tensor. However, [10] uses SIFT for tensor representation. One of the ingredients of the success of our method is the use of rich feature descriptors in the tensorial representation of the video. We also define a canonical correlation kernel that seamlessly integrates the advantages of lower dimensional representation of videos with a discriminative classifier like SVM. This also enables us to weigh the features differently for further improving the recognition performance.

Wolf and Shashua [31] had extended the notion of canonical correlation analysis (CCA) to introduce a kernelized version of the same in KCCA. This is considerably different from what we do in this paper. Rather than kernelizing CCA, we are interested in defining a kernel which can be used in many situations where canonical correlation is used. However, we find that the correlation analysis with the help of KCCA, by computing similarities over projections on nonlinear manifolds, also provides useful information for action recognition. We simultaneously consider the correlations computed over linear as well as nonlinear manifolds of the video data tensor. While the individual correlation coefficients can not be used as a valid measure for comparing two action videos in a kernel setting, their sum becomes a valid measure. Our kernel is simple to compute and visualize, starting from a canonical correlation analysis. However this enables a host of useful tricks. (i) we can use SVM along with CCA/KCCA based feature extraction (ii) we can simultaneously compute similarities over multiple features in a single framework (iii) we can optimally fuse the advantages of the bag of words representation, (as in [29]) and tensor based methods, (as in [9, 16]) for action recognition. Multiple feature kernels are often combined using multiple kernel learning (MKL) [17] framework. MKL techniques [30] have been used in action recognition for combining different contextual features. Our kernel yields superior results with simple (say pixel values) features, it allows to compare videos without hand coding or tracking. An illustration of the framework of the proposed method is shown in Figure 1. The method has 3 steps, in the first step a given video is represented using a 3D tensor. In the second step this 3D tensor is decomposed into three 2D tensors/matrices. Finally, for given two videos, CCK is computed from their corresponding 2D tensors, which involves CCA and KCCA. This results in a set of correlation coefficients. Sum of these correlations gives the CCK for the given 2 videos.

Our approach is experimented on four popular action video data sets. Our single feature representation (with pixel values as features) outperforms most

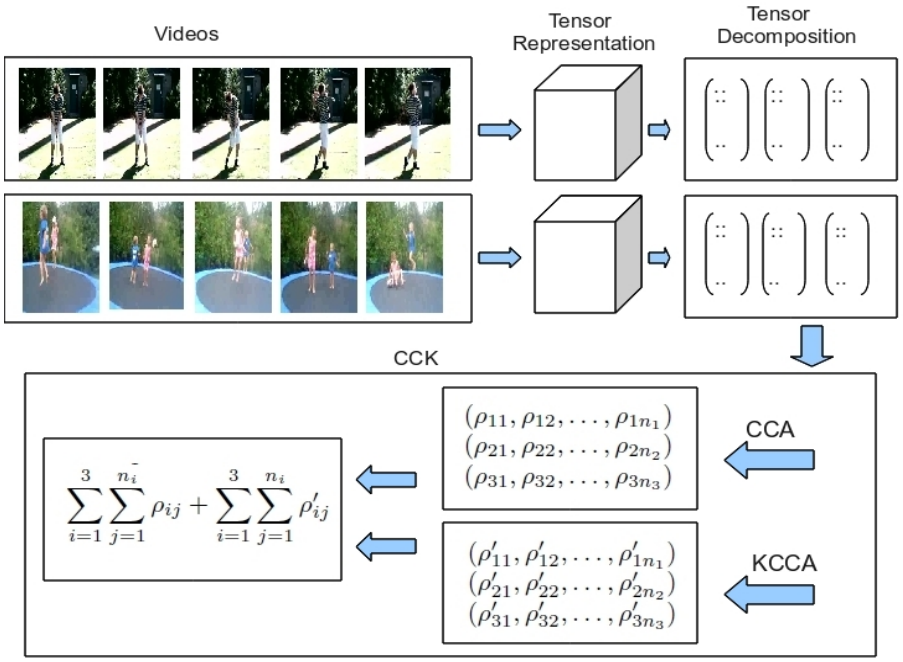

Fig. 1: An illustration of the proposed method. In the first step, videos are represented using 3D tensors. Next, these 3D tensors are flattened to obtain three 2D matrices. Finally, CCK is computed as the sum of correlations between the flattened matrices obtained from CCA and KCCA.

competitive methods, which use more powerful features. We also show that recognition performance further improves by intuitive and seamless fusion of multiple feature kernels. Our proposed canonical correlation kernel is explained in Section 3. Experimental results are discussed in detail in Section 4.

2 Related Work

A wide spectrum of features and representations has been used for action recognition in the past. Initial attempts like Motion Intensity/History Images represented the whole video as a single image and used traditional feature extraction for recognizing actions. Such features typically captured global motion information in a compact manner. On the other extreme, local information captured using features like SIFT [25], HOG [8, 28], LBP [11] and MBH were also used for describing video frames and found to be useful for action recognition. The need for defining a set of distinct descriptors for video (from images) was realized, and many features like SIFT and HOG got extended to video by defining them over a volume rather than over a 2D grid [25]. However, a successful direction

has been to track the features over frames and to compute the descriptors over this track to represent the action content [21, 22]. By making these tracks denser, Wang *et al.* [28] obtained excellent results on popular data sets. We argue that such dense and feature rich representations can result in superior results when used along with the learned representations from video volumes.

A video can be represented as a third order data tensor denoted as $\mathcal{V} \in \mathbb{R}^{X \times Y \times Z}$. Where, X and Y are spatial dimensions, which gives spatial information and Z is a time axis, which gives temporal information of a video. This representation is bulky and noisy for deriving effective representations for action recognition. A wide variety of decomposition techniques [9, 15, 16] were used for deriving lower dimensional representations from these data tensors. A successful method [9] is to start with projections of these tensors over multiple dimensions resulting in matrices and deriving algorithms that work on these matrices. Often matrix projections on the spatial and temporal axes are represented as points on a manifold [15]. Canonical correlation analysis represents an action video with the help of a vector of discriminatively selected subset of correlation coefficients defined over a linear manifold [2]. The kernalized version of CCA [31] measures the correlation on a nonlinear implicit manifold. Alternate techniques for representing the action videos include those based on tangent bundles [15], product manifolds [16] etc. In tangent bundles [15], data tensors are factorized to a set of tangent spaces on a Grassmann manifold. In product manifold [16], each tensor is considered as a point on a product manifold and the tensor is factorized using a modified High Order Singular Value Decomposition. Naoki *et al.* [1] represent the gait dynamics trained from multiple training videos by a standard manifold.

Once the video is represented in a lower dimension, a similarity measure is defined to compare two videos. This similarity/dissimilarity measure could be simple Euclidean or cosine similarity as in canonical correlation. Our canonical correlation kernel is based on the principal angles between the points on a manifold [2] resulting from the tensor decomposition of two videos. This is a simple, yet powerful generalization of the similarity computation in a canonical correlation analysis.

Canonical correlation analysis [2] has been successfully applied for image set comparison in robust object recognition, and later extended for action recognition [9, 15]. It gives the linear relationships between two set of random variables (or two matrices) in terms of correlations. Given two matrices, CCA finds two projections one for each of the matrices such that the correlation between projections of the matrices is maximized. The correlation constants (correlations) between the projections of the matrices gives a similarity measure between original matrices. For any given two matrices, its canonical correlations can be computed in two ways. One is from the singular values of given matrices and other is using the eigendecomposition of the matrix which is obtained by pre and post multiplying the cross-covariance matrix by the inverse square root of the covariance matrix of given matrices. In this paper, we have followed the SVD based solution [2]. For the matrices P and Q of dimension $n \times d_1$ and $n \times d_2$ ($n > min\{d_1, d_2\}$), we denote their orthonormal matrices as $\bar{P}$ and $\bar{Q}$ of dimen-

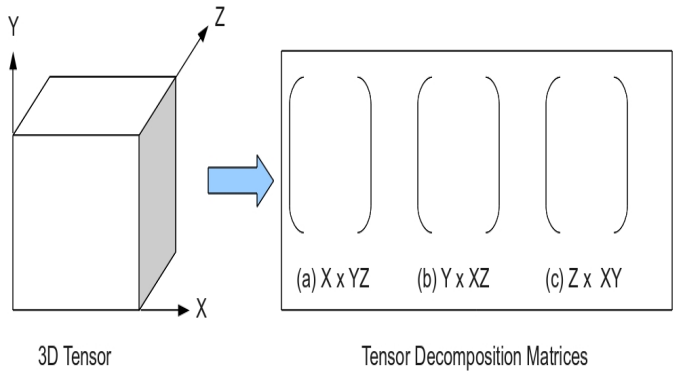

Fig. 2: Flattening of Tensor representation. A third order tensor is flattened into 3 second order tensors (2D matrices) X×YZ, Y×XZ and Z×XY. Here, X, Y are spatial axes and Z is a time axis. The first tensor matrix (X×YZ) (a) is obtained by keeping X fixed and flattening YZ into a single dimension. The other decomposition matrices are also obtained in a similar way.

sion $n \times rank(P)$ and $n \times rank(Q)$. The d correlation constants $(\rho_1, \rho_2, \ldots, \rho_d)$, where $d = min\{rank(P), rank(Q)\}$, for the matrices P and Q, can be computed as the singular values of the matrix $\bar{P}^T\bar{Q}$.

Our definition of the canonical correlation kernel is motivated by the need to use tensorial representation framework along with a discriminative classifier like SVM. Diverse set of features can also be used for tensorial representation.

3 Canonical Correlation Kernels

We start with a tensor representation of the video volume. To begin with, let us consider that elements of the tensor are the pixel (or intensity) values. However this representation can easily scale to other dense representations, where more powerful feature descriptors (e.g. SIFT) are used to encode the local information. Our objective is to define an effective similarity measure that can scale for multiple features. We achieve this with the help of a canonical correlation kernel defined based on the canonical correlation analysis [2], which has already been used in action recognition [9]. Our kernel is far more effective than the discriminatively selected (using boosting) correlation coefficients used in [9] for comparing videos as can be seen in Section 4.

Use of CCA for defining a kernel for the action recognition task is motivated by multiple factors: (i) Canonical correlation allows us to define a kernel, which can be used in a maximum margin discriminative framework like SVM, and used for seamless combination of multiple features and representations. For example, in Section 4, we show that Bag of Words based methods can be used along with the tensor based ones (ii) TCCA based method had shown some success in recognizing actions in the past [9]. (iii) similarity of the videos can be computed

by projecting them over a linear manifold in CCA and nonlinear manifold in KCCA in the same framework. (iv) correlation coefficients measures the similarity in a more intuitive manner compared to the popular distance functions like Euclidean. We find this to be very effective for comparing videos. Note that the principal angle based similarity computation (cosine similarity) is widely used in text domain, and has proven to be more appropriate compared to the L^p norms in a wide range of problems.

3.1 Canonical Correlation Kernel (CCK)

Given two random vectors $\mathbf{x}$ and $\mathbf{y}$, canonical correlation analysis measures the similarity by finding the correlation of these two vectors after a set of linear transformations. Assume $\mathbf{x}$ gets linearly transformed with $\mathbf{u}$ as $\mathbf{x}' = \mathbf{u}^t\mathbf{x}$ and $\mathbf{y}$ as $\mathbf{y}' = \mathbf{v}^t\mathbf{y}$, then canonical correlation is defined as the maximum possible correlation over *all* possible transformations $\mathbf{u}$ and $\mathbf{v}$. For a video recognition problem, this implies that the video is getting transformed (or features are getting extracted) such that the correlation in the most appropriate feature space is maximized. Thus, we simultaneously learn the most appropriate features and the similarity in that feature space. The method remains same irrespective of whether $\mathbf{x}$ and $\mathbf{y}$ are vectors, matrices or tensors. Correlation coefficients ρ_i measures the similarity in the projected space as the cosine of the angles between the linear manifold.

A limitation of the above is the use of linear transformation while extracting features. Wolf and Shashua [31] addressed this problem by kernelizing the canonical correlation by defining the transformation in an implicit feature space. For example, a kernel $\kappa(\mathbf{x}, \mathbf{u}) = \phi(\mathbf{x})^T\phi(\mathbf{u})$ does the feature extraction by finding a manifold instead of a linear subspace. This generalizes the classical canonical correlation and provides a mechanism for extracting a richer set of features. In our implementation, we use an exponential kernel (for KCCA) as $\kappa(\mathbf{x}, \mathbf{y}) = e^{-\beta d(\mathbf{x},\mathbf{y})}$. Here also, similarity is measured using correlation coefficients ρ_i' computed for nonlinear manifolds. For our experiments, we use the similarities computed over both linear as well as nonlinear manifolds i.e., ρ_i and ρ_i', simultaneously.

Our video representation is essentially a third order tensor. While working with pixel values we scale the video to a smaller size as explained in the experimental section. While using feature representations (eg. SIFT descriptors), we sample the image/frame further and compute the features at a smaller set of grid points. A tensor thus obtained from pixel values or feature descriptors is first flattened to obtain a matrix. Here, we use three kinds of flattening corresponding to spatial and time axis. This is done similar to many of the previous works [9, 15, 16]. This flattening is explained in Figure 2. For a given video of size $l \times m \times n$, where n is the number of frames and $l \times m$ is the size of each frame, the flattening corresponding to time axis is leads to a matrix of size $d_1 \times n$. Each column in the matrix corresponds to a frame of the video and the number of columns is same as the number of frames in the video/tensor. Here d_1 is the length of the feature descriptor obtained from each frame, for pixel values this

is equal to $l \cdot m$. Computation of feature descriptors is explained in Section 3.2. Similarities between two videos is then computed with the help of canonical correlations coefficients between the corresponding flattened matrices, which are basically the principal angles between the subspaces.

Given two videos $\mathcal{V}_1$ and $\mathcal{V}_2$ denote their i^{th} $(i = 1, 2, 3)$ flatten matrices as V_1^i, V_2^i, we define the canonical correlation kernel corresponding to the i^{th} flatten matrices as the sum of all the correlation coefficients obtained from both CCA and KCCA over V_1^i and V_2^i. i.e.,

$$K^{'}(V_1^i, V_2^i) = \sum_{j=1}^{d} \rho_j + \sum_{j=1}^{d} \rho_j' \tag{1}$$

where, $d = min\{rank(V_1^i), rank(V_2^i)\}$ and ρ_i, ρ_i' are the correlation coefficients obtained from CCA and KCCA over V_1^i and V_2^i. We flatten the third order tensor video into three second order matrices (tensor). Our final canonical correlation kernel between two videos is the sum of canonical correlation kernels obtained from three flattening matrices. i.e.,

$$CCK(\mathcal{V}_1, \mathcal{V}_2) = \sum_{i=1}^{3} K^{'}(V_1^i, V_2^i) \tag{2}$$

3.2 Feature Kernels

CCK can be computed for pixel values as well as for other extracted features. In both cases the procedure of computation remains the same. For every feature, we first compute its feature matrices corresponding to three flattenings then CCK is computed over these matrices. For a given feature, its feature matrix corresponding to time axis flattening over a video of n frames is an $d \times n$ matrix. The i^{th} column in this matrix represents the feature descriptor obtained from the i^{th} frame and d is the descriptor length. In addition to the pixel values, we use the features SIFT [18], HOG (histograms of oriented gradients) [5], MBH (motion boundary histogram) [6] and HOF (histograms of optical flow) [14]. Among these descriptors, HOG and HOF have shown to give good results for action recognition [29]. HOG captures static local information where as HOF captures motion information. Dalal *et al.* [6] proposed MBH for human detection, it captures the relative motion between the pixels. In our experiments, these feature descriptors are extracted as follows,

- For **SIFT**, we divide a frame into a fixed grids, where size of each grid is 4×4, and SIFT is extracted at each grid location. Final descriptor for the corresponding frame is obtained by the concatenation of all the SIFT descriptors.
- For **HOG**, we compute HOG descriptor using a window of size 4×4 and a bin size of 9. Concatenation of all the local histograms is taken as the final representation.

- For **MBH**, similar to SIFT, we divide a frame into grids of size 4×4 and MBH is computed at each of the grid locations. We take bin size of 8 and patch size of 32.
- For **HOF**, we divide the frame into grids of size 6×6 and HOF descriptors are computed at each of these grid locations with a neighborhood size of 24×24 and concatenation of all these descriptors is taken as the final feature descriptor. Here, we quantize the orientations into 9 bins.

All the descriptors are normalized to zero mean and unit variance.

CCK is reasonably insensitive to various factors such as temporal misalignment, scale variations and background variations. Temporal misalignment comes from the affine invariance property of CCA. Since CCK is computed after the normalizing the videos, scale variation is also taken care of. When background changes significantly, insensitivity depends on the features used. In our experience, CCK defined over HOF and MBH is practically insensitive to this.

3.3 Classifier:

We use a support vector machine (SVM) classifier. Feature kernels are combined [7] by giving equal weights to all the kernels or by giving high weight to one kernel and zero weightage to all other kernels. One can also use multiple kernel learning (MKL) [4] for combing the feature kernels. If $\kappa_j(\cdot,\cdot)$ is the canonical correlation kernel computed using j^{th} descriptor, then the final kernel is obtained as the linear combination of all the kernels

$$\kappa(\mathbf{x},\mathbf{y}) = \sum_j d_j \kappa_j(\mathbf{x},\mathbf{y}) \tag{3}$$

If all the d_js are equal then the final kernel κ will be the simple average of given kernels. In all our experiments, we use libsvm [3] package for the SVM classifier.

4 Experiments

In this section, we present a detailed evaluation of our proposed kernel (CCK). CCK is based on canonical correlation analysis and it can be used to compare the videos for action recognition. We evaluate various components of the proposed kernel to justify our choices. We compare our results with the previously published works.

We report our results on four publicly available standard action datasets: Cambridge gesture, KTH human action, Youtube and UCF sports action datasets. Some sample frames from the datasets are shown in Figure 3. For multiclass classification, in all our experiments we use a one-vs-rest SVM classifier and select the class with the highest score. We perform the experiments on raw video representation using pixel values as well as on feature representations. For combining

Fig. 3: Some sample frames from video sequences on Cambridge Gesture (first row), KTH (second row), UCF (third row) and Youtube (fourth row) datasets. For cambridge, background is uniform in most of the sequences. For KTH, background is homogenous and static. UCF and youtube videos have non uniform background. Youtube dataset has large variations in camera motion

the feature representations, we use the simple weighted scheme as discussed in Section 3.3. In all our experiments, the kernel matrices obtained over the given datasets are positive definite. For the given four datasets, average accuracy over all the classes is reported as the performance measure.

4.1 Datasets and Experimental Setting

Cambridge Gesture Dataset: The cambridge gesture dataset [12][1] consists of 900 video sequences belonging to 9 action classes. These videos are captured using 5 different lighting conditions from two subjects. The data is divided into five sets (one for each illumination setting), where each set contains a total of

[1] ftp://mi.eng.cam.ac.uk/pub/CamGesData

180 video sequences. we use 10 random videos from each class in set5 (plain illumination setting) for training and all videos from the remaining sets (set1, set2, set3 and set4) for testing as reported in [9, 15, 16]. For our experimental setup, we use tensors of size 20×20×20, where 20 frames from each sequence were obtained by uniform sampling. Each frame in the sequence is resized to 20×20. Classifier is trained using the randomly chosen 90 videos. Accuracies are reported by taking the average over 10 trials.

KTH Dataset: The KTH dataset [26][2] contains 600 videos from 6 human action classes. In most of the videos background is homogeneous and static. Each type of action is performed by 25 different actors in indoor and outdoor settings. We extract the human actions by following the procedure used in [15, 16]. Our tensor formulation is identical to [12, 15] by constructing tensors of size 20×20×32. Experiments are carried out using leave one out cross validation, which is performed by dividing the dataset into 25 folds (each fold containing 24 videos of the same person).

UCF Sports Action Dataset: The UCF sports action dataset [24][3] contains 150 videos from 10 different sports action classes. The number of videos for each class varies from 6 to 22. This dataset has large intra-class variability. Similar to [15], we use tensors of size 32×32×64 where each frame is resized to 32×32. Similar to the KTH dataset, experiments are performed using leave-one-out cross validation, where each video is taken as a separate fold and the remaining 149 videos are used for training.

Youtube Dataset: The youtube dataset [19][4] contains 1168 videos from 11 different classes. It is one of the challenging datasets. This is mainly due to the presence of significant camera motion, viewpoint transitions, varying illumination conditions and cluttered backgrounds in the videos. Videos for each class are divided into 25 folds based on the persons performing that action. We use tensors of size 32×32×64, where each frame is resized to 32×32. Accuracies are reported using leave one out cross validation over the predefined 25 folds.

4.2 Results and Discussions

The methods, which we compare with our proposed method can be divided into two categories. Methods which uses tensor decomposition representation for videos [9, 15, 16] and which uses feature representations [8, 20, 28].

CCK on individual Features We report the individual feature accuracies using CCK in Table 1. We compare the results with dense trajectories [28] feature kernels, which are computed using χ^2 kernel [14] and kernels are combined in a multichannel approach similar to [27]. CCK perform better than the dense trajectories for HOG and HOF over all the four datasets. This indicates the strength of CCK, the superiority comes from the temporal context it embeds

[2] http://www.nada.kth.se/cvap/actions/
[3] http://www.cs.ucf.edu/vision/public_html
[4] http://www.cs.ucf.edu/liujg/YouTube_Action_dataset.html

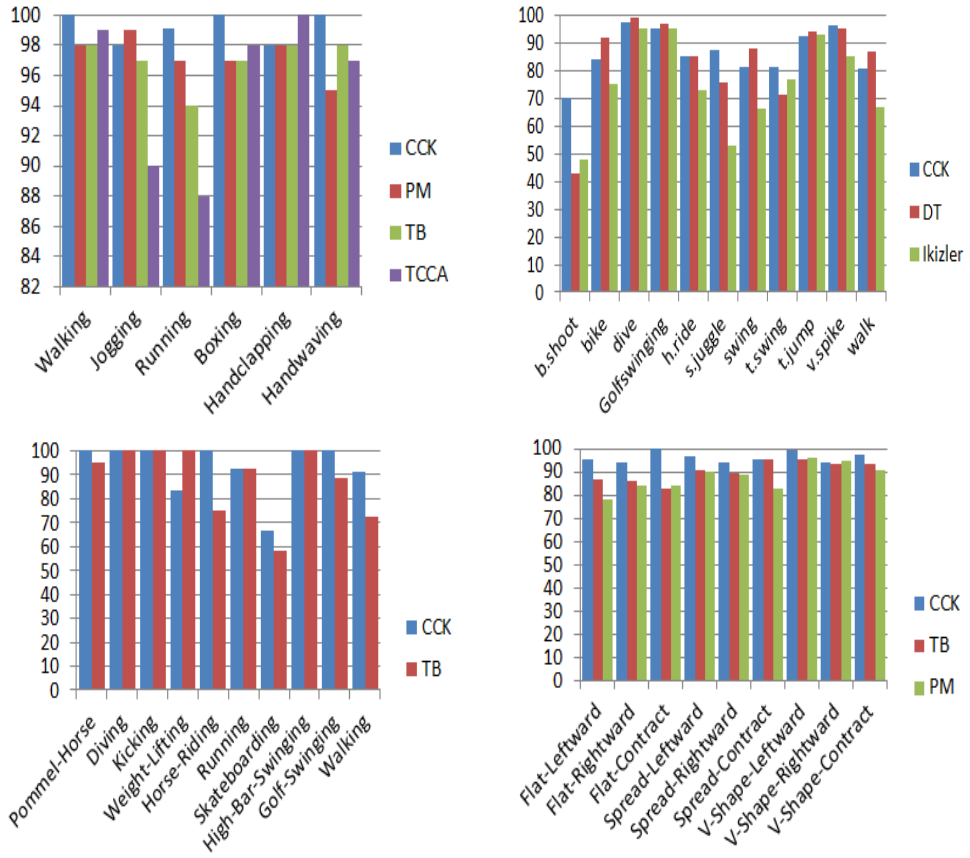

Fig. 4: Classwise accuracies on KTH and Youtube (in first row), UCF and Cambridge (in second row). For KTH, we compare with Tangent Bundle (TB) [15], Product Manifold (PM) [16] and TCCA [9]. For Youtube, we compare with Dense Trajectories (DT) [28] and Ikizler [8]. For UCF, we compare with Tangent Bundle (TB) [15]. For Cambridge, we compare with Tangent Bundle (TB) [15] and Product Manifold (PM) [16].

from the videos. Thus, it best suits for the action recognition task over other kernels.

CCK using Pixel values Pixel value accuracies are reported in Table 2. Using pixel values alone, we achieve a significant improvement of 2.1% and 5.3% for Cambridge and UCF datasets over previous work. We report 97.5% on KTH dataset, which is an improvement of 0.5% over [15]. The CCK using pixel values is comparable to state-of-the-art [28] on Youtube dataset. This indicates that pixel values alone are good enough for CCK to get the better results. The main reason behind this is that, for tensorial representation, the actions in the videos

	Cambridge		UCF		KTH		Youtube	
	CCK	DT [28]	CCK	DT [28]	CCK	DT [28]	CCK	DT [28]
Pixel Values	93.1	-	93.5	-	97.5	-	82.5	-
HOG	89.0	-	**83.8**	**83.8**	**98.3**	86.5	**83.2**	74.5
SIFT	95.1	-	85.7	-	98.6	-	79.1	-
HOF	95.2	-	**81.5**	77.6	**94.3**	93.2	**80.4**	72.8
MBH	75.1	-	80.4	**84.8**	**98.9**	95.0	80.1	**83.9**
Combined	96.4	-	**93.5**	88.2	**98.9**	94.2	**86.3**	84.2

Table 1: Comparison of our proposed canonical correlation kernel (CCK) with DT (Dense trajectories) [28] over different feature descriptors on Cambridge, UCF, KTH and Youtube datasets. Results are reported on pixel values, HOG, SIFT, HOF and MBH. For each dataset, we report average accuracy over all the classes. Final accuracies after combining the features are also displayed. Dense trajectories [28] have not used pixel values and SIFT. Accuracies are reported in %.

Method	Cambridge	UCF	KTH	Youtube
TCCA [9]	82±3.5	-	95.33	-
Product Manifold [16]	88	-	97	-
Tangent Bundle [15]	91	88	97	-
Dense trajectories [28]	-	88.2	94.2	84.2
Le *et al.* [20]	-	86.5	93.9	75.8
Ikizler-Cinbis *et al.* [8]	-	-	-	75.21
Jiang Wang *et al.* [30]	-	-	93.8	-
Proposed (Using pixel values)	93.1	**93.5**	97.5	82.5
Proposed (Using multiple features)	96.4	93.5	**98.9**	86.3
Proposed (CCK feature kernels + DT feature kernels)	**97.2**	93.5	98.9	**86.6**

Table 2: Comparison of our proposed method with other state-of-the-art methods. Here, we give the accuracy of our proposed kernel (CCK) over simple pixel values and using multiple features (pixel values, HOG, SIFT and MBH). Accuracies over multiple features are obtained using simple weighting scheme. Finally, combined accuracy using CCK feature kernels and of DT [28] feature kernels are also reported. Accuracies are reported in %.

are well represented using pixel values compared to other features. This gives the superiority of CCK.

CCK using multiple features We combine the feature descriptors to further enhance the accuracy. Features are combined using simple weighted scheme as discussed in Section 3.3. We report the combined feature accuracies using weighted scheme in Table 2, it compares our results with the previous methods. We achieve an improvement of 5.4%, 5.3%, 1.9% and 2.1% over Cambridge, UCF, KTH and Youtube datasets. This indicates that videos can be well represented using multiple features in a tensorial representation framework. One can also use MKL [13] for combining the feature descriptors.

Accuracy over the combination of canonical correlation feature kernels and DT (dense trajectory) [28] feature kernels are shown in Table 2. It further improved the accuracy over all the datasets. This indicates that CCK can be easily integrated with other features such as Bag of words histograms to achieve further improvement in the accuracy. We also compare the classwise accuracies for all the datasets with other methods in Figure 4. On KTH, CCK gives the best results for 5 out of 6 action classes, as compared to [16]. On Youtube, our method got best results over 5 classes compared to [28]. For UCF, we got best results for 9 out of 10 classes and for Cambridge, we got best results for all the classes compared to the state-of-the-art.

In summary, our superiority comes from (1) The proposed CCK which embeds temporal context in the videos into similarity measure (2) Seamless fusion of multiple features into tensor representation.

5 Conclusions

In this paper, we have introduced the canonical correlation kernel (CCK), which enables comparison of videos in a kernel framework. This kernel function works well for action recognition as it embeds the temporal context in the videos. We have also shown that multiple features can be seamlessly integrated into CCK to further enhance the recognition performance. We hope that our work open up scope for a class of action recognition algorithms which use, tensor representation, multiple feature description and use a discriminatively max margin classification.

Acknowledgement. G Nagendar is supported under TCS research fellowship scheme.

References

1. Akae, N., Mansur, A., Makihara, Y., Yagi, Y.: Video from nearly still: an application to low frame-rate gait recognition. In: CVPR. (2012) 1537–1543
2. Bjorck, A., Golub, G.H.: Numerical methods for computing angles between linear subspaces. Mathematics of Computation, 27(123) (1973) 579–594
3. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology (2011) 27:1–27:27 Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
4. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning (2002) 131–159
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In:CVPR (1). (2005) 886–893
6. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: ECCV (2). (2006) 428–441
7. Gehler, P., Nowozin, S.: Infinite kernel learning. In: Proceedings of NIPS 2008 Workshop on "Kernel Learning: Automatic Selection of Optimal Kernels". (2008)
8. Ikizler-Cinbis, N., Sclaroff, S.: Object, scene and actions: Combining multiple features for human action recognition. In: ECCV (1). (2010) 494–507

9. Kim, T.K.,Wong, S.F., Cipolla, R.: Tensor canonical correlation analysis for action classification. In: CVPR. (2007)
10. Kim, T.K., Cipolla, R.: Gesture recognition under small sample size. In: ACCV (1). (2007) 335–344
11. Kellokumpu, V., Zhao, G., Pietikainen, M.: Human activity recognition using a dynamic texture based method. In: BMVC. (2008)
12. Kim, T.K., Cipolla, R.: Canonical correlation analysis of video volume tensors for action categorization and detection. IEEE Trans. Pattern Anal. Mach. Intell. (2009) 1415–1428
13. Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Muller, K.R., Zien, A.: Efficient and accurate lp-norm multiple kernel learning. In: NIPS. (2009) 997–1005
14. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008)
15. Lui, Y.M., Beveridge, J.R.: Tangent bundle for human action recognition. In: FG. (2011) 97–102
16. Lui, Y.M., Beveridge, J.R., Kirby, M.: Action classification on product manifolds. In: CVPR. (2010) 833–839
17. Lanckriet, G.R.G., Cristianini, N., Bartlett, P.L., Ghaoui, L.E., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research (2004) 27–72
18. Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. (1999) 1150–1157
19. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos. In: CVPR. (2009) 1996–2003
20. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatiotemporal features for action recognition with independent subspace analysis. In:CVPR. (2011) 3361–3368
21. Messing, R., Pal, C., Kautz, H.A.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV. (2009) 104–111
22. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: Action recognition through the motion analysis of tracked features. In: Workshop on Video-Oriented Object and Event Classification. ICCV (2009)
23. Nowak, E., F.J., Triggs, B.: Sampling strategies for bag-of-features image classification. In: ECCV. (2006)
24. Rodriguez, M., J.A., Shah., M.: Action mach: A spatiotemporal maximum average correlation height filter for action recognition. In: In CVPR. (2008)
25. Sun, J., Wu, X., Yan, S., Cheong, L.F., Chua, T.S., Li, J.: Hierarchical spatiotemporal context modeling for action recognition. In: CVPR. (2009) 2004–2011
26. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: ICPR. (2004) 32–36
27. Ullah, M.M., Parizi, S.N., Laptev, I.: Improving bag-of-features action recognition with non-local cues. In: BMVC. (2010) 1–11
28. Wang, H., Klaser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR. (2011) 3169–3176
29. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC. (2009)
30. Wang, J., Chen, Z., Wu, Y.: Action recognition with multiscale spatio-temporal contexts. In: CVPR. (2011) 3185–3192
31. Wolf, L., Shashua, A.: Kernel principal angles for classification machines with applications to image sequence interpretation. In: CVPR. (2003) 635–642