Joint Recognition and Segmentation of Actions via Probabilistic Integration of Spatio-Temporal Fisher Vectors

Johanna Carvajal[1,3], Chris McCool[2], Brian Lovell[1],
and Conrad Sanderson[1,3,4](✉)

[1] University of Queensland, Brisbane, Australia
[2] Queensland University of Technology, Brisbane, Australia
[3] NICTA, Brisbane, Australia
[4] Data61, CSIRO, Brisbane, Australia
conrad.sanderson@nicta.com.au

Abstract. We propose a hierarchical approach to multi-action recognition that performs joint classification and segmentation. A given video (containing several consecutive actions) is processed via a sequence of overlapping temporal windows. Each frame in a temporal window is represented through selective low-level spatio-temporal features which efficiently capture relevant local dynamics. Features from each window are represented as a Fisher vector, which captures first and second order statistics. Instead of directly classifying each Fisher vector, it is converted into a vector of class probabilities. The final classification decision for each frame is then obtained by integrating the class probabilities at the frame level, which exploits the overlapping of the temporal windows. Experiments were performed on two datasets: s-KTH (a stitched version of the KTH dataset to simulate multi-actions), and the challenging CMU-MMAC dataset. On s-KTH, the proposed approach achieves an accuracy of 85.0 %, significantly outperforming two recent approaches based on GMMs and HMMs which obtained 78.3 % and 71.2 %, respectively. On CMU-MMAC, the proposed approach achieves an accuracy of 40.9 %, outperforming the GMM and HMM approaches which obtained 33.7 % and 38.4 %, respectively. Furthermore, the proposed system is on average 40 times faster than the GMM based approach.

1 Introduction

Research on human action recognition can be divided into two areas: **(i)** single-action recognition, and **(ii)** multi-action recognition. In most computer vision literature, action recognition approaches have concentrated on single actions, where each video to be classified contains only one action. However, when observing realistic human behaviour in natural settings, the fundamental problem is segmenting and recognising actions from a *multi-action* sequence [1]. It is challenging due to the high variability of appearances, shapes, possible occlusions, large variability in the temporal scale and periodicity of human actions, the complexity of articulated motion, the exponential nature of all possible movement combinations, as well as the prevalence of irrelevant background [2,3].

H. Cao et al. (Eds.): PAKDD 2016 Workshops, LNAI 9794, pp. 115–127, 2016.
DOI: 10.1007/978-3-319-42996-0_10

Hoai et al. [2] address joint segmentation and classification by classifying temporal regions using a multi-class Support Vector Machine (SVM) and performing segmentation using dynamic programming. A similar approach is presented in [4], where the temporal relationship between actions is considered. Borzeshi et al. [5] proposed the use of hidden Markov models (HMMs) with irregular observations (termed HMM-MIO) to perform multi-action recognition. More recently, Carvajal et al. [6] proposed to model each action using a Gaussian mixture model (GMM) approach where classification decisions are made using overlapping temporal windows. A drawback of [2,4,5] is that they have a large number of parameters to optimise. Furthermore, [5] requires an extra stage to reduce dimensionality due to use of very high dimensional feature vectors, while [2,4] require fully labelled annotations for training. One downside of [6] is that for each action and scenario, a model with a large number of Gaussians is required, making the system computationally expensive.

Typically, the aforementioned approaches used for the multi-action recognition task can be classified as either generative or discriminative models. The approaches presented in [5,6] are generative models, while those presented in [2,4] are discriminative models. Generative and discriminative models have complementary strengths. Generative models can easily deal with variable length sequences and missing data, while also being easier to design and implement [7,8]. In contrast, discriminative models often achieve superior classification and generalisation performance [7,8]. An ideal recognition system would hence combine these two separate but complementary approaches.

The Fisher vector (FV) approach [7,9,10] allows for the fusion of both generative and discriminative models. In contrast to the popular Bag of Words (BoW) approach [11] which describes images by histograms of visual words, the FV approach describes images by deviations from a probabilistic visual vocabulary model. The resulting vectors can then be used by an SVM for final classification. Recently, FV has been successfully applied to the single-action recognition problem [12,13].

A reliable low-level feature descriptor is a crucial stage for the success of an action recognition system. One popular descriptor for action recognition is Spatio-Temporal Interest Points (STIPs) [14]. However, STIP based descriptors have several drawbacks [15,16]: they are computationally expensive, unstable, imprecise and can result in unnecessarily sparse detections. See Fig. 1 for a demonstration of STIP based detection. Other feature extraction techniques used for action recognition include gradients [6] and optical flow [16,17]. Each pixel in the gradient image helps extract relevant information, eg. edges (see Fig. 1). Since the task of action recognition is based on a sequence of frames, optical flow provides an efficient way of capturing the local dynamics [16].

Contributions. In this paper we propose a novel hierarchical system to perform multi-action segmentation and recognition. A given video is processed as a sequence of overlapping temporal windows. Each frame in a temporal window is represented through selective low-level spatio-temporal features, based on a combination of gradients with optical flow. Interesting features from each

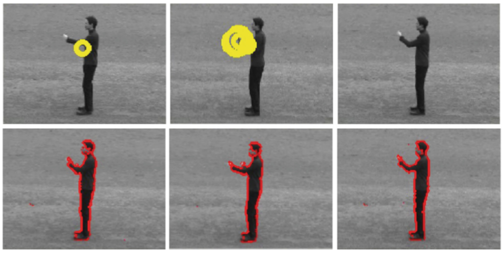


Fig. 1. Top row: feature extraction based on Spatio-Temporal Interest Points (STIPs) is often unstable, imprecise and overly sparse. Bottom row: interest pixels (marked in red) obtained using magnitude of gradient. (Color figure online)

temporal window are then pooled and processed to form a Fisher vector. Instead of directly classifying each Fisher vector, a multi-class SVM is employed to generate a vector of probabilities, with one probability per class. The final classification decision (action label) for each frame is then obtained by integrating the class probabilities at the frame level, which exploits the overlapping of the temporal windows. The proposed system hence combines the benefits of generative and discriminative models.

To the best of our knowledge, the combination of probabilistic integration with Fisher vectors is novel for the *multi-action* segmentation and recognition problem. In contrast to [2,4,5], the proposed system requires fewer parameters to be optimised. We also avoid the need for a custom dynamic programming definition as in [2,4]. Lastly, unlike the GMM approach proposed in [6] the proposed method requires only one GMM for all actions, making it considerably more efficient.

2 Proposed Method

The proposed system has a hierarchical nature, stemming from progressive reduction and transformation of information, starting at the pixel level. The system is comprised of four main components:

1. Division of a given video into overlapping multi-frame temporal windows, followed by extracting interesting low-level spatio-temporal features from each frame in each window.
2. Pooling of the interesting features from each temporal window to generate a sequence of Fisher vectors.
3. Conversion of each Fisher vector into a vector of class probabilities with the aid of a multi-class SVM.
4. Integration of the class probabilities at the frame level, leading to the final classification decision (action label) for each frame.

Each of the components is explained in more detail in the following subsections.

2.1 Overlapping and Selective Feature Extraction

A video $\mathcal{V} = (\boldsymbol{I}_t)_{t=1}^{T}$ is an ordered set of T frames. We divide $\mathcal{V}$ into a set of overlapping temporal windows $(\mathcal{W}_s)_{s=1}^{S}$, with each window having a length of L frames. To achieve overlapping, the start of each window is one frame after the start of the preceding window. Each temporal window is hence defined as a set of frame identifiers: $\mathcal{W}_s = (t_{\text{start}},\ \ldots,\ t_{\text{start}-1+L})$.

Each frame $\boldsymbol{I}_t \in \mathbb{R}^{r \times c}$ can be represented by a set of feature vectors $F_t = \{\boldsymbol{f}_p\}_{p=1}^{N_t}$ (with $N_t < r \cdot c$) corresponding to interesting pixels. Following [6], we first extract the following $D = 14$ dimensional feature vector for each pixel in a given frame t:

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

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

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

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

The first four gradient-based features in Eq. (2) represent the first and second order intensity gradients at pixel location (x, y). The last two gradient features represent gradient magnitude and gradient orientation. The optical flow based features in Eq. (3) represent in order: the horizontal and vertical components of the flow vector, the first order derivatives with respect to time, the divergence and vorticity of optical flow [17].

Typically only a subset of the pixels in a frame correspond to the object of interest. As such, we are only interested in pixels with a gradient magnitude greater than a threshold τ [18]. We discard feature vectors from locations with a small magnitude. In other words, only feature vectors corresponding to interesting pixels are kept. This typically results in a variable number of feature vectors per frame. See the bottom part in Fig. 1 for an example of the retained pixels.

2.2 Representing Windows as Fisher Vectors

Given a set of feature vectors, the Fisher Vector approach encodes the deviations from a probabilistic visual dictionary, which is typically a diagonal GMM. The parameters of a GMM with K components can be expressed as $\{w_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k\}_{k=1}^{K}$, where, w_k is the weight, $\boldsymbol{\mu}_k$ is the mean vector, and $\boldsymbol{\sigma}_k$ is the diagonal covariance matrix for the k-th Gaussian. The parameters are learned via the Expectation Maximisation algorithm [19] on training data.

For each temporal window $\mathcal{W}_s$, the feature vectors are pooled into set X containing $N = \sum_{t \in \mathcal{W}_s} N_t$ vectors. The deviations from the GMM are then accumulated via [10]:

$$\mathcal{G}_{\boldsymbol{\mu}_k}^{X} = \frac{1}{N\sqrt{w_k}} \sum\nolimits_{n=1}^{N} \gamma_n(k) \left(\frac{\boldsymbol{f}_n - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right) \tag{4}$$

$$\mathcal{G}_{\boldsymbol{\sigma}_k}^{X} = \frac{1}{N\sqrt{2w_k}} \sum\nolimits_{n=1}^{N} \gamma_n(k) \left[\frac{(\boldsymbol{f}_n - \boldsymbol{\mu}_{\boldsymbol{k}})^2}{\boldsymbol{\sigma}_k^2} - 1 \right] \tag{5}$$

where vector division indicates element-wise division and $\gamma_n(k)$ is the posterior probability of $\boldsymbol{f}_n$ for the k-th component: $\gamma_n(k) = w_k \mathcal{N}(\boldsymbol{f}_n|\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k) / \left(\sum_{i=1}^{K} w_i \mathcal{N}(\boldsymbol{f}_n|\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)\right)$.

The Fisher vector for window $\mathcal{W}_s$ is represented as the concatenation of $\mathcal{G}^X_{\boldsymbol{\mu}_k}$ and $\mathcal{G}^X_{\boldsymbol{\sigma}_k}$ (for $k = 1, \ldots, K$) into vector $\boldsymbol{\Phi}_s$. As $\mathcal{G}^X_{\boldsymbol{\mu}_k}$ and $\mathcal{G}^X_{\boldsymbol{\sigma}_k}$ are D-dimensional, $\boldsymbol{\Phi}_s$ has the dimensionality of $2DK$. Note that we have omitted the deviations for the weights as they add little information [10].

2.3 Generation of Probability Vectors

For each Fisher vector we generate a vector of probabilities, with one probability per action class. First, a multi-class SVM [20] is used to predict class labels, outputting a set of raw scores. The scores are then transformed into a probability distribution over classes by applying Platt scaling [21]. The final probability vector derived from Fisher vector $\boldsymbol{\Phi}_s$ is expressed as:

$$\boldsymbol{q}_s = [P(l = 1|\boldsymbol{\Phi}_s), \ \cdots, \ P(l = A|\boldsymbol{\Phi}_s)] \tag{6}$$

where l indicates an action class label and A is the number of action classes. The parameters for the multi-class SVM are learned using Fisher vectors obtained from pre-segmented actions in training data.

2.4 Integrating Probability Vectors to Label Frames

As the temporal windows are overlapping, each frame is present in several temporal windows. We exploit the overlapping to integrate the class probabilities at the frame level. The total contribution of the probability vectors to each frame t is calculated by:

$$\boldsymbol{Q}_t = \sum\nolimits_{s=1}^{S} \mathbf{1}_{\mathcal{W}_s}(t) \cdot \boldsymbol{q}_s \tag{7}$$

where $\mathbf{1}_{\mathcal{W}_s}(t)$ is an indicator function, resulting in 1 if $t \in \mathcal{W}_s$, and 0 otherwise. The estimated action label for frame t is then calculated as:

$$\widehat{l}_t = \underset{l=1,\ldots,A}{\arg\max} \ \boldsymbol{Q}_t^{[l]} \tag{8}$$

where $\boldsymbol{Q}_t^{[l]}$ indicates the l-th element of $\boldsymbol{Q}_t$.

3 Experiments

We evaluated our proposed method for joint action segmentation and recognition on two datasets: **(i)** a stitched version of the KTH dataset [22], and **(ii)** the challenging Carnegie Mellon University Multi-Modal Activity Dataset (CMU-MMAC) [23]. The results are reported in terms of frame-level accuracy as the ratio between the number of matched frames over the total number of frames.

The s-KTH (stitched KTH) dataset is obtained by simply concatenating existing single-action instances into sequences [5]. The KTH dataset contains 25 subjects performing 6 types of human actions and 4 scenarios. The scenarios vary from indoor, outdoor, scale variations, and different clothes. Each original video of the KTH dataset [22] contains an individual performing the same action. This action is performed four times and each subdivision or action-instance (in terms of start-frame and end-frame) is provided as part of the dataset. This dataset contains 2391 action-instances, with a length between 1 and 14 s [24]. The image size is 160×120 pixels, and temporal resolution is 25 frames per second.

The action-instances (each video contains four instances of the action) were picked randomly, alternating between the two groups of {boxing, hand-waving, hand-clapping} and {walking, jogging, running} to accentuate action boundaries. See Fig. 2 for an example. The dataset was divided into two sets as in [5,6]: one for training and one for testing. In total, 64 and 36 multi-action videos were used for training and testing, respectively. We used 3-fold cross-validation.

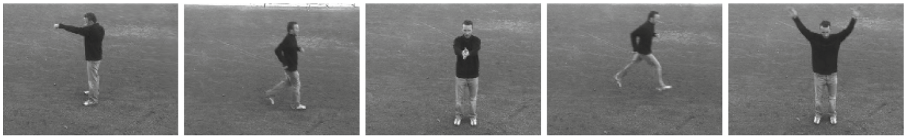


Fig. 2. Example of a multi-action sequence in the stitched version of the KTH dataset (s-KTH): boxing, jogging, hand clapping, running, hand waving and walking.

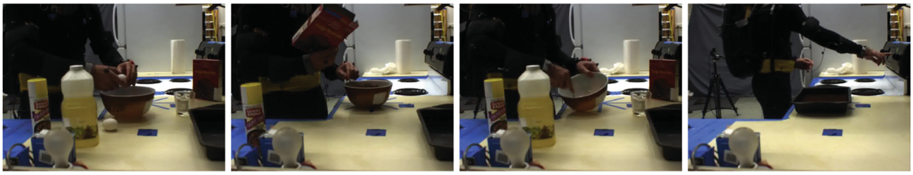


Fig. 3. Example of a challenging multi-action sequence in the CMU-MMAC kitchen dataset: crack, read, stir, and switch-on.

The CMU-MMAC dataset is considerably more challenging as it contains realistic multi-action videos [23]. A kitchen was built to record subjects preparing and cooking food according to five recipes. This dataset has occlusions, a cluttered background, and many distractors such as objects being deliberately moved. For our experiments we have used the same subset as per [5], which contains 12 subjects making brownies. The subjects were asked to make brownies in a natural way (no instructions were given). Each subject making the brownie is partially seen, as shown in Fig. 3.

The videos have a high resolution and are longer than in s-KTH. The image size is 1024×768 pixels, and temporal resolution is 30 frames per second. The average duration of a video is approximately 15,000 frames and the average length of an action instance is approximately 230 frames (7.7 s), with a minimum length of 3 frames (0.1 s) and a maximum length of 3,269 frames (108 s) [5]. The dataset was annotated using 14 labels, including the actions *close*, *crack*, *open*, *pour*, *put*, *read*, *spray*, *stir*, *switch-on*, *take*, *twist-off*, *twist-on*, *walk*, and the remaining actions (eg. frames in between two distinct actions) were grouped under the label *none* [25]. We used 12-fold cross-validation, using one subject for testing on a rotating basis.

All videos were converted into gray-scale. Additionally, the videos from the CMU-MMAC dataset were re-scaled to 128×96 to reduce computational requirements. Based on [6] and preliminary experiments on both datasets, we used $\tau = 40$, where τ is the threshold used for selection of interesting low-level feature vectors (Sect. 2.1). Although the interesting feature vectors are calculated in all frames, we only use the feature vectors extracted from every second frame in order to speed up processing.

Parameters for the visual vocabulary (GMM) were learned using a large set of descriptors obtained from training videos. Specifically, we randomly sampled 100,000 feature vectors for each action and then pooled all the resultant feature vectors from all actions for training. Experiments were performed with three separate GMMs with varying number of components: $K = \{64, 128, 256\}$. We have not evaluated larger values of K due to increased computational complexity and hence the exorbitant amount of time required to process the large CMU-MMAC dataset.

To learn the parameters of the multi-class SVM, we used video segments containing single actions. For s-KTH this process is straightforward as the videos have been previously segmented. The CMU-MMAC dataset contains continuous multi-actions. For this reason, to train our system we obtain one Fisher vector per action in each video, using the low-level feature vectors belonging to that specific action.

3.1 Effect of Window Length and Dictionary Size

We have evaluated the performance of two variants of the proposed system: **(1)** probabilistic integration with Fisher vectors (**PI-FV**), and **(2)** probabilistic integration with BoW histograms (**PI-BoW**), where the Fisher vector representation is replaced with BoW representation. We start our experiments by studying the influence of the segment length L, expressed in terms of seconds. The results are reported in Figs. 4 and 5, in terms of average accuracy over the folds.

Using the PI-FV variant (Fig. 4), we found that using $L = 1s$ and $K = 256$ leads to the best performance on the s-KTH dataset. For the CMU-MMAC dataset, the best performance is obtained with $L = 2.5\,s$ and $K = 64$. Note that using larger values of K (128 and 256) leads to worse performance. We attribute this to the large variability of appearance in the dataset, where the training data may not be a good representative of test data. As such, using a large value of K may lead to overfitting to the training data.

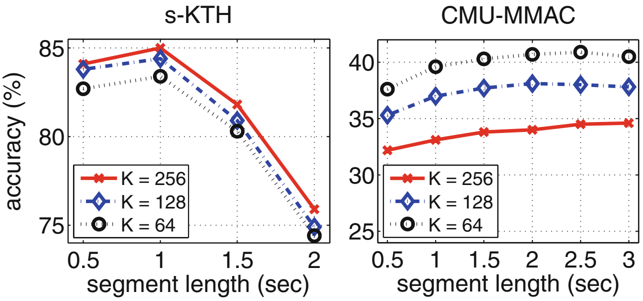


Fig. 4. Performance of the proposed **PI-FV** approach for varying the segment length on the s-KTH and CMU-MMAC datasets, in terms of average frame-level accuracy over the folds.

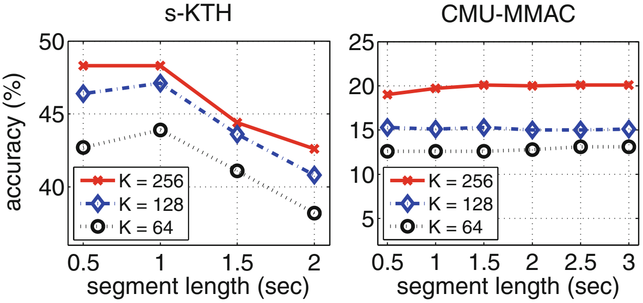


Fig. 5. As per Fig. 4, but showing the performance of the **PI-BoW** variant (where the Fisher vector representation is replaced with BoW representation).

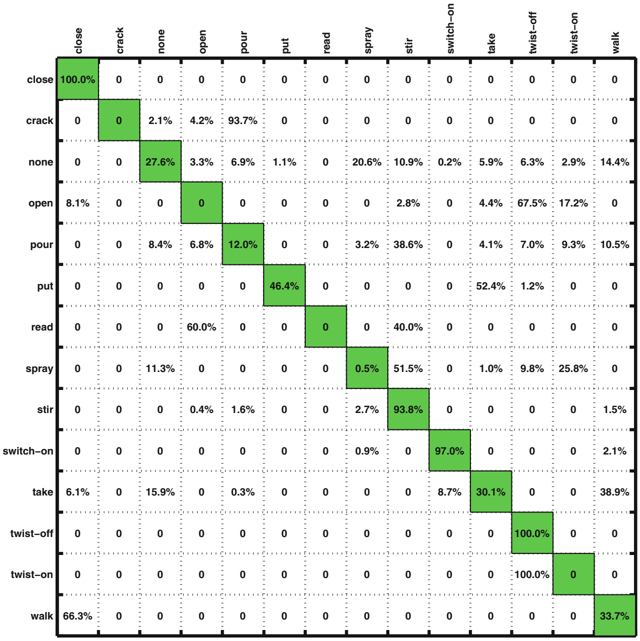
The optimal segment length for each dataset is different. We attribute this to the s-KTH dataset containing short videos whose duration is between $1s$ and $7s$, while CMU-MMAC has a large range of action durations between $0.1s$ and $108s$. While the optimal values of L and K differ across the datasets, the results also show that relatively good overall performance across both datasets can be obtained with $L = 1s$ and $K = 64$.

The results for the PI-BoW variant are shown in Fig. 5. The best performance for the PI-BoW variant on the s-KTH dataset is obtained using $L = 1s$ and $K = 256$, while on the CMU-MMAC dataset it is obtained with $L = 2.5s$ and $K = 256$. These are the same values of L and K as for the PI-FV variant. However, the performance of the PI-BoW variant is consistently worse than the PI-FV variant on both datasets. This can be attributed to the better representation power of FV. Note that the visual dictionary size K for BoW is usually higher in order to achieve performance similar to FV. However, due to the large size of the CMU-MMAC dataset, and for direct comparison purposes, we have used the same range of K values throughout the experiments.

3.2 Confusion Matrices

Figures 6 and 7 show the confusion matrices for the PI-FV and PI-BoW variants on the CMU-MMAC dataset. The confusion matrices show that in 50 % of the cases (actions), the PI-BoW variant is unable to recognise the correct action. Furthermore, the PI-FV variant on average obtains better action segmentation than PI-BoW.

For five actions (*crack*, *open*, *read*, *spray*, *twist-on*), PI-FV has accuracies of 0.5 % or lower. Action *crack* implies crack and pour eggs into a bowl, but it's annotated only as *crack*, leading to confusion between *crack* and *pour*. We suspect that actions *read* and *spray* are poorly modelled due to lack of training data; they are performed by a reduced number of subjects. Action *twist-on* is confused with *twist-off* which are essentially the same action.



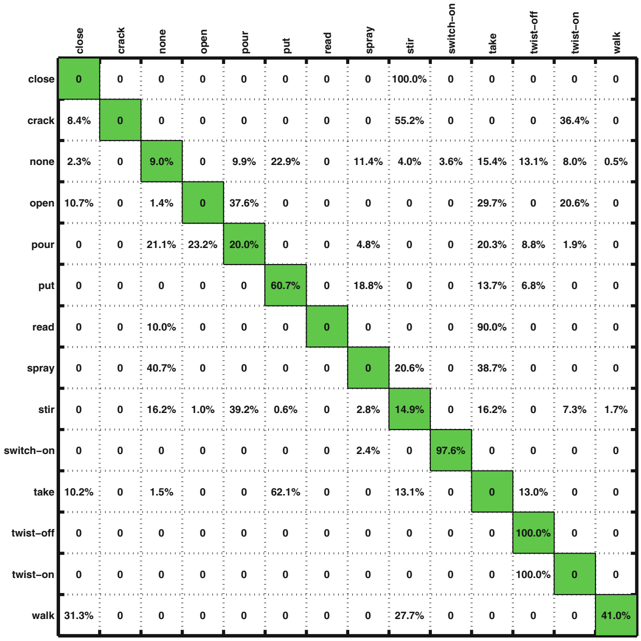
	close	crack	none	open	pour	put	read	spray	stir	switch–on	take	twist–off	twist–on	walk
close	100.0%	0	0	0	0	0	0	0	0	0	0	0	0	0
crack	0	0	2.1%	4.2%	93.7%	0	0	0	0	0	0	0	0	0
none	0	0	27.6%	3.3%	6.9%	1.1%	0	20.6%	10.9%	0.2%	5.9%	6.3%	2.9%	14.4%
open	8.1%	0	0	0	0	0	0	0	2.8%	0	4.4%	67.5%	17.2%	0
pour	0	0	8.4%	6.8%	12.0%	0	0	3.2%	38.6%	0	4.1%	7.0%	9.3%	10.5%
put	0	0	0	0	0	46.4%	0	0	0	0	52.4%	1.2%	0	0
read	0	0	0	60.0%	0	0	0	0	40.0%	0	0	0	0	0
spray	0	0	11.3%	0	0	0	0	0.5%	51.5%	0	1.0%	9.8%	25.8%	0
stir	0	0	0	0.4%	1.6%	0	0	2.7%	93.8%	0	0	0	0	1.5%
switch–on	0	0	0	0	0	0	0	0.9%	0	97.0%	0	0	0	2.1%
take	6.1%	0	15.9%	0	0.3%	0	0	0	0	8.7%	30.1%	0	0	38.9%
twist–off	0	0	0	0	0	0	0	0	0	0	0	100.0%	0	0
twist–on	0	0	0	0	0	0	0	0	0	0	0	100.0%	0	0
walk	66.3%	0	0	0	0	0	0	0	0	0	0	0	0	33.7%

Fig. 6. Confusion matrix for the PI-FV variant on the CMU-MMAC dataset.

3.3 Comparison with GMM and HMM-MIO

We have compared the performance of the PI-FV and PI-BoW variants against the HMM-MIO [5] and stochastic modelling [6] approaches previously used for multi-action recognition. The comparative results are shown in Table 1.

The proposed PI-FV method obtains the highest accuracy of 85.0 % and 40.9 % for the s-KTH and CMU-MMAC datasets, respectively. In addition to higher accuracy, the proposed method has other advantages over previous techniques. There is just one global GMM (representing the visual vocabulary).



	close	crack	none	open	pour	put	read	spray	stir	switch–on	take	twist–off	twist–on	walk
close	0	0	0	0	0	0	0	0	100.0%	0	0	0	0	0
crack	8.4%	0	0	0	0	0	0	0	55.2%	0	0	0	36.4%	0
none	2.3%	0	9.0%	0	9.9%	22.9%	0	11.4%	4.0%	3.6%	15.4%	13.1%	8.0%	0.5%
open	10.7%	0	1.4%	0	37.6%	0	0	0	0	0	29.7%	0	20.6%	0
pour	0	0	21.1%	23.2%	20.0%	0	0	4.8%	0	0	20.3%	8.8%	1.9%	0
put	0	0	0	0	0	60.7%	0	18.8%	0	0	13.7%	6.8%	0	0
read	0	0	10.0%	0	0	0	0	0	0	0	90.0%	0	0	0
spray	0	0	40.7%	0	0	0	0	0	20.6%	0	38.7%	0	0	0
stir	0	0	16.2%	1.0%	39.2%	0.6%	0	2.8%	14.9%	0	16.2%	0	7.3%	1.7%
switch–on	0	0	0	0	0	0	0	2.4%	0	97.6%	0	0	0	0
take	10.2%	0	1.5%	0	0	62.1%	0	0	13.1%	0	0	13.0%	0	0
twist–off	0	0	0	0	0	0	0	0	0	0	0	100.0%	0	0
twist–on	0	0	0	0	0	0	0	0	0	0	0	100.0%	0	0
walk	31.3%	0	0	0	0	0	0	0	27.7%	0	0	0	0	41.0%

Fig. 7. As per Fig. 6, but using the PI-BoW variant.

This is in contrast to [6] which uses one GMM (with a large number of components) for each action, leading to high computational complexity. The HMM-MIO method in [5] requires the search for many optimal parameters, whereas the proposed method has just two parameters (L and K).

Table 1. Comparison of the proposed methods (PI-FV and PI-BoW) against several recent approaches on the stitched version of the KTH dataset (s-KTH) and the challenging CMU-MMAC dataset.

Method	**s-KTH**	**CMU-MMAC**
HMM-MIO [5]	71.2 %	38.4 %
Stochastic modelling [6]	78.3 %	33.7 %
PI-FV	**85.0%**	**40.9%**
PI-BoW	48.0 %	20.1 %

3.4 Wall-Clock Time

Lastly, we provide an analysis of the computational cost (in terms of wall-clock time) of our system and the stochastic modelling approach. The wall-clock time is measured under optimal configuration for each system, using a Linux machine with an Intel Core processor running at 2.83 GHz.

On the s-KTH dataset, the stochastic modelling system takes on average 228.4 min to segment and recognise a multi-action video. In comparison, the proposed system takes 5.6 min, which is approximately 40 times faster.

4 Conclusions and Future Work

In this paper we have proposed a hierarchical approach to multi-action recognition that performs joint segmentation and classification in videos. Videos are processed through overlapping temporal windows. Each frame in a temporal window is represented using selective low-level spatio-temporal features which efficiently capture relevant local dynamics and do not suffer from the instability and imprecision exhibited by STIP descriptors [14]. Features from each window are represented as a Fisher vector, which captures the first and second order statistics. Rather than directly classifying each Fisher vector, it is converted into a vector of class probabilities. The final classification decision for each frame (action label) is then obtained by integrating the class probabilities at the frame level, which exploits the overlapping of the temporal windows.

The proposed approach has a lower number of free parameters than previous methods which use dynamic programming or HMMs [5]. It is also considerably less computationally demanding compared to modelling each action directly with a GMM [6].

Experiments were done on two datasets: s-KTH (a stitched version of the KTH dataset to simulate multi-actions), and the more challenging CMU-MMAC dataset (containing realistic multi-action videos of food preparation). On s-KTH, the proposed approach achieves an accuracy of 85.0 %, considerably outperforming two recent approaches based on GMMs and HMMs which obtained 78.3 % and 71.2 %, respectively. On CMU-MMAC, the proposed approach achieves an accuracy of 40.9 %, outperforming the GMM and HMM approaches which obtained 33.7 % and 38.4 %, respectively. Furthermore, the proposed system is on average 40 times faster than the GMM based approach.

Possible future areas of exploration include the use of the fast Fisher vector variant proposed in [26], where for each sample the deviations for only one Gaussian are calculated. This can deliver a large speed up in computation, at the cost of a small drop in accuracy [27].

Acknowledgements. NICTA is funded by the Australian Government through the Department of Communications, as well as the Australian Research Council through the ICT Centre of Excellence program.

References

1. Buchsbaum, D., Canini, K.R., Griffiths, T.: Segmenting and recognizing human action using low-level video features. In: Annual Conference of the Cognitive Science Society (2011)

2. Hoai, M., Lan, Z.Z., De la Torre, F.: Joint segmentation and classification of human actions in video. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3265–3272 (2011)
3. Shi, Q., Wang, L., Cheng, L., Smola, A.: Discriminative human action segmentation and recognition using semi-Markov model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2008)
4. Cheng, Y., Fan, Q., Pankanti, S., Choudhary, A.: Temporal sequence modeling for video event detection. In: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2235–2242 (2014)
5. Borzeshi, E., Perez Concha, O., Xu, R., Piccardi, M.: Joint action segmentation and classification by an extended hidden Markov model. IEEE Sig. Process. Lett. **20**, 1207–1210 (2013)
6. Carvajal, J., Sanderson, C., McCool, C., Lovell, B.C.: Multi-action recognition via stochastic modelling of optical flow and gradients. In: Workshop on Machine Learning for Sensory Data Analysis (MLSDA), pp. 19–24. ACM (2014). http://dx.doi.org/10.1145/2689746.2689748
7. Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. Adv. Neural Inf. Process. Syst. **11**, 487–493 (1998)
8. Lasserre, J., Bishop, C.M.: Generative or discriminative? Getting the best of both worlds. In: Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., West, M. (eds.) Bayesian Statistics, vol. 8, pp. 3–24. Oxford University Press, Oxford (2007)
9. Csurka, G., Perronnin, F.: Fisher vectors: beyond bag-of-visual-words image representations. In: Richard, P., Braz, J. (eds.) VISIGRAPP 2010. CCIS, vol. 229, pp. 28–42. Springer, Heidelberg (2011)
10. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the Fisher vector: theory and practice. Int. J. Comput. Vis. **105**, 222–245 (2013)
11. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. Br. Mach. Vis. Conf. (BMVC) **124**(1–124), 11 (2009)
12. Oneata, D., Verbeek, J., Schmid, C.: Action and event recognition with Fisher vectors on a compact feature set. In: International Conference on Computer Vision (ICCV), pp. 1817–1824 (2013)
13. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: International Conference on Computer Vision (ICCV) (2013)
14. Laptev, I.: On space-time interest points. Int. J. Comput. Vis. **64**, 107–123 (2005)
15. Cao, L., Tian, Y., Liu, Z., Yao, B., Zhang, Z., Huang, T.: Action detection using multiple spatial-temporal interest point features. In: International Conference on Multimedia and Expo (ICME), pp. 340–345 (2010)
16. Kliper-Gross, O., Gurovich, Y., Hassner, T., Wolf, L.: Motion interchange patterns for action recognition in unconstrained videos. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 256–269. Springer, Heidelberg (2012)
17. Ali, S., Shah, M.: Human action recognition in videos using kinematic features and multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **32**, 288–303 (2010)
18. Guo, K., Ishwar, P., Konrad, J.: Action recognition from video using feature covariance matrices. IEEE Trans. Image Process. **22**, 2479–2494 (2013)
19. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)

20. Crammer, K., Singer, Y.: On the algorithmic implementation of multiclass kernel-based vector machines. J. Mach. Learn. Res. **2**, 265–292 (2001)
21. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Adv. Large Margin Classifiers **10**, 61–74 (1999)
22. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local SVM approach. Int. Conf. Pattern Recogn. (ICPR) **3**, 32–36 (2004)
23. De la Torre, F., Hodgins, J.K., Montano, J., Valcarcel, S.: Detailed human data acquisition of kitchen activities: the CMU-multimodal activity database (CMU-MMAC). In: CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research (2009)
24. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Sequential deep learning for human action recognition. In: Salah, A.A., Lepri, B. (eds.) HBU 2011. LNCS, vol. 7065, pp. 29–39. Springer, Heidelberg (2011)
25. Spriggs, E.H., Torre, F.D.L., Hebert, M.: Temporal segmentation and activity classification from first-person sensing. In: IEEE Workshop on Egocentric Vision, CVPR (2009)
26. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep Fisher networks for large-scale image classification. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) Advances in Neural Information Processing Systems, vol. 26, pp. 163–171 (2013)
27. Parkhi, O.M., Simonyan, K., Vedaldi, A., Zisserman, A.: A compact and discriminative face track descriptor. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2014)