Computer Vision and Image Understanding 000 (2016) 1–15

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Understanding and localizing activities from correspondences of clustered trajectories[☆]

Francesco Turchini, Lorenzo Seidenari[*], Alberto Del Bimbo

Università degli Studi di Firenze, MICC, Florence, Italy

ARTICLE INFO

Article history:
Received 21 March 2016
Revised 15 November 2016
Accepted 29 November 2016
Available online xxx

Keywords:
Action recognition
Action localization
Sport analytics

ABSTRACT

We present an approach for human activity recognition based on trajectory grouping. Our representation allows to perform partial matching between videos obtaining a robust similarity measure. This approach is extremely useful in sport videos where multiple entities are involved in the activities. Many existing works perform person detection, tracking and often require camera calibration in order to extract motion and imagery of every player and object in the scene. In this work we overcome this limitations and propose an approach that exploits the spatio-temporal structure of a video, grouping local spatio-temporal features unsupervisedly. Our robust representation allows to measure video similarity making correspondences among arbitrary patterns. We show how our clusters can be used to generate frame-wise action proposals. We exploit proposals to improve our representation further for localization and recognition. We test our method on sport specific and generic activity dataset reporting results above the existing state-of-the-art.

1. Introduction

Human activity recognition is a fundamental problem in computer vision (Karaman et al., 2014; Oneata et al., 2013; Wang and Schmid, 2013) with many applications such as video retrieval (Revaud et al., 2013), automatic visual surveillance (Kosmopoulos et al., 2012; Roshtkhari and Levine, 2013; Ryoo, 2011) and human computer interaction (Wang et al., 2012). Sports represent one of the most viewed content on digital tv and on the web. Sports are watched by millions of people and broadcasters are constantly improving user experience by providing real-time statistics of games.

Recently, many computer vision researchers directed their efforts in the automatic analysis of sports videos. Sports video analytics is often performed to collect statistics on player positions during games, extracting individual trajectories and team formation patterns (Atmosukarto et al., 2013; Hsu et al., 2014; Liu et al., 2013).

Some commercial systems are available and used to track players http://www.stats.com/sportvu/sportvu-basketball-media or the ball(http://www.hawkeyeinnovations.co.uk/sports/tennis). These expensive systems are often targeted to a better enforcement of rules, which may become challenging in sports with high speed moving objects such as Tennis (http://www.hawkeyeinnovations.co.uk/sports/tennis, 2016). Player tracking can generate statistics that can be fed into player public databases to increase web site visitors among casual fans and sport enthusiasts.

There is little or no development of industrial grade algorithms for single camera generic sport activity understanding. We believe this to be an important direction to investigate since there are many interesting and valuable tasks to be solved.

Classifying player actions in sports is an extremely relevant task that can provide several commercial and professional applications. Speakers, analysts and directors may obtain in real-time similar plays from the current or other games providing an improved experience for the audience. Head coaches may easily classify all the plays of a certain player to track improvement or to analyse other teams tactics; finally gameplay statistics can be automatically gathered such as the amount of shots on goal and corner kicks a soccer team had in a game or a season.

Many action recognition datasets are comprised of just sport videos and there is interest in recognizing sports as concepts in videos (Karpathy et al., 2014; Niebles et al., 2010; Soomro and Zamir, 2014). More effort has been poured in the analysis of team tactics and activity (Bialkowski et al., 2013; Gade and Moeslund, 2013). Team activities are best defined by player positions in the field, for this reason many works exploit this datum. Many methods are based on multi-camera systems deployed to get full coverage of the court.

[☆] An earlier version of this paper appeared in a conference proceeding (Turchini et al., 2015).

[*] Corresponding author.
E-mail address: lorenzo.seidenari@unifi.it (L. Seidenari).

http://dx.doi.org/10.1016/j.cviu.2016.11.007
1077-3142/© 2016 Elsevier Inc. All rights reserved.

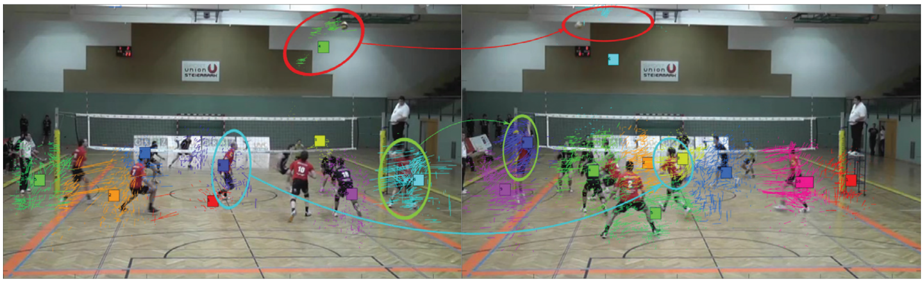


Fig. 1. Example of cluster matching between two videos from the same class of the volley dataset. The cluster with features generated by the ball in motion is correctly matched as well as the ones with players approaching the net.

There are few methods, apart from generic action recognition systems, that attempt to classify player activities without localizing and tracking individual players (Ballan et al., 2010). Indeed several techniques require a calibrated fixed view to fuse visual with geometrical features such as player trajectories or positions in the field.

In this paper we propose an activity recognition method that targets complex activities with possibly multiple individuals involved, which are typical of team sports. Differently from previously published work on sport activity recognition, our method does not require calibrated views of the field, player track annotations or player tracking, neither is based on player team recognition. Our method automatically groups visual features forming a robust representation of videos. The main idea behind our approach is shown in Fig. 1.

We base our approach on improved trajectories (IDT) (Wang et al., 2013) and we do not encode explicitly player positions or the temporal sequence of a video. We automatically group trajectories and define a match kernel able to make arbitrary correspondences of spatio-temporal patterns.

Our method is similar to Karaman et al. (2014), Gaidon et al. (2013) but differently from Gaidon et al. (2013) we do not require a hierarchical partitioning of the features. Nor we have the requirement of using quantized local features that have worst performance with respect to Fisher encoded descriptors. Compared to Karaman et al. (2014) we do not use pooling importance maps. Karaman et al.obtain a spatio-temporal scene decomposition by processing Hierachical Space-Time Segments (Ma et al., 2013) while we just rely on our feature grouping method. Therefore we have a less strict requirement on feature pooling, and encoding allowing for better local feature representation. Furthermore, we have a more general approach to infer the spatio-temporal structure of the video with respect to Karaman et al. (2014), not relying on object tracking or segmentation.

The flexibility and generality of the proposed approach requires the encoding of multiple high dimensional feature vectors per video. This has the drawback of increasing the spatial complexity with respect to other single signature methods. Nevertheless we show how to cope with this issue, compressing our vectors and defining a quantized version of our algorithm that allows to deal with larger datasets with little loss in classification accuracy.

We also show how our clusters can be used as per-frame action proposals with a two-fold benefit: we use the proposals to localize actions in space and time and we derive an additional powerful representation based on convolutional networks that is naturally plugged into our framework.

We test our method on two sport activity datasets, improving accuracy with respect to previously published methods by a large margin. We also show state-of-the-art results on UCF-Sports, Hollywood2 and HighFive showing that our method is also a viable generic action recognition system.

1.1. Related work

We briefly review some recent contributions on automatic sport activity recognition. Atmosukarto et al. (2013) developed a method to recognize offensive team formation in American football. Their method applies robust video stitching and exploits the localization of the line of scrimmage to compute a feature based on gradient intensity on the offensive side of the line. Bialkowski et al. (2013) avoid tracking players but apply player detection and team recognition. The method exploits multiple calibrated views of the field to locate players in the field. Team activity is recognized computing team field occupancy maps.

Ballan et al. (2010) match videos using a kernel for sequences derived from the Needleman–Wunch distance (NWD). The temporal structure of a video is a fundamental cue for recognizing complex events such as sport activities. Their approach is based on the fact that similar actions should share similar appearance in a similar sequence. The main limitation of their method is the use of static features (SIFT) and the fact that NWD is not designed to make arbitrary correspondences between sequences. Brun et al. (2016) propose a similar approach, computing a fast global alignment kernel, exploiting frame based depth features.

Waltner et al. (2014) propose a method to recognize individual player activities in volleyball. Their method exploits player detection and camera calibration. Single player activities are recognized using a boosting based approach learning from static and motion local features. They also compute a contextual feature based on player position for which they require player team recognition.

Most of the information needed to train effective discriminative classifiers for actions resides in motion. Local motion features were first proposed by Laptev et al. (2008) and named spatio-temporal interest points (STIP). The STIP algorithm is an extension for videos of local image feature detection and description. After identifying multi-scale regions, multiple local descriptors, based on histograms of optical flow and gradient orientation are computed. Wang et al. (2009) evaluated several sampling strategies and local descriptors showing that avoiding feature detection in favour of dense exhaustive sampling improves the results. However, in a more recent line of research, using local feature tracking as a mean of sampling and to extract better descriptors prevailed (Jain et al., 2013; Jiang et al., 2012; Raptis et al., 2012; Wang et al., 2015a). The idea behind trajectory based sampling is to compute the final local feature aligning the local frame, thus obtaining a stabilized version of the local pattern. To recover the motion information trajectory geometry can be used as a feature itself (Wang et al., 2013).

A sensible feature tracking quality improvement is obtained with camera motion compensation (Jain et al., 2013; Wang et al., 2015a). Once dominant motion is extracted it is possible to extract only the relevant objects that are moving. Trajectories can therefore effectively discard background features and static objects.

Please cite this article as: F. Turchini et al., Understanding and localizing activities from correspondences of clustered trajectories, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2016.11.007

Trajectory estimation can be improved using a warped version of the optical flow, aligning subsequent frames with a transformation (Jain et al., 2013; Wang et al., 2015a). Wang et al. propose to increase the accuracy of the transformation estimate using a person detector to remove trajectories that are less likely to be generated by the camera motion. This approach is suitable for edited videos such as movies where actors fill at least half of the frame area.

Vrigkas et al. (2014), model videos using a GMM over optical flow features. Action recognition is then performed matching GMM fitted on a test video with GMMs learned for every action class at training time. A limitation of this approach is the requirement of bounding box annotation to learn GMMs on training data. Kviatkovsky et al. (2014) propose a descriptor based on covariances of optical flow and silhouettes, that is extremely fast and is suitable for human computer interaction scenarios.

Most of the generic action recognition methods represent the video as a global entity, pooling a high cardinality feature set using either some classical coding algorithm like bag-of-words (Chatfield et al., 2011) or more modern ones like VLAD (Jégou et al., 2010) or Fisher Vectors (Sánchez et al., 2013). Feature coding methods that perform better on image classification and retrieval have proved the most effective also on action recognition (Oneata et al., 2013; Wang et al., 2015a).

Following the success in object classification, deep convolutional neural networks (DCNN) have been applied to action recognition (Ravanbakhsh et al., 2015; Tran et al., 2015; Wang et al., 2015b). Although the results from aforementioned approaches are promising, it has been reported that they do not always outperform handcrafted features, especially on harder datasets such as Hollywood2 (Bruni et al., 2016).

A different line of research attempted to exploit trajectories as a mean of gaining insight on the spatio-temporal structure of the video (Gaidon et al., 2013; Karaman et al., 2014; Raptis et al., 2012). Gaidon et al. (2013) show that hierarchically partitioning a video using a top-down procedure improves the representation with respect to a global feature pooling. They represent the video imposing a hierarchy and compare videos with a tree matching algorithm using hard quantized features. Raptis et al. (2012) group trajectories and use a MRF to formulate the subgraph matching used to compare videos. Finally Karaman et al. (2014) exploit clustered Hierarchical Space Time Segments (HSTS)(Ma et al., 2013) to generate feature pooling maps and compare videos with an efficient graph matching algorithm (Feragen et al., 2013).

Action classification algorithms are not usually able to provide the performer location in space-time. Action localization is the task of predicting the spatio-temporal volume in which an action takes place. Localization can be addressed as a supervised (Lan et al., 2011; Tian et al., 2013) or as unsupervised (Ma et al., 2013; Yu and Yuan, 2015) task. Supervised methods train action detectors discriminatively, provided performers bounding boxes. Tian et al. (2013) extend the Deformable Parts Model (DPM) to locate spatio-temporal objects, i.e. actions, formulating the Spatio-Temporal Deformable Part Models (DSDPM). Lan et al. (2011) provide a different approach, treating person location as a latent variable and inferring it simultaneously with the action class. Jain et al. (2014) extend Selective Search (Uijlings et al., 2013) to the temporal domain finding hierarchical segmentation of supervoxels. Wang et al. (2014) propose an approach which is similar, in principle, to Tian et al., but with a stronger supervision. They use a structural SVM formulation to search for body parts. This approach requires body joint annotations.

Some methods, specifically devised for action localization, exploit the temporal coherence of the video (Gkioxari and Malik, 2015; Weinzaepfel et al., 2015). Weinzaepfel et al. (2015) performs an exhaustive search to refine EdgeBoxes proposal and combine an instance-level and a class-level classifier to track proposal over a video. Gkioxari et al. have a very close approach to Weinzaepfel et al. (2015), learning CNN specific action classifiers to generate spatio-temporal volumes, named “tubes”. To refine the result they employ a global optimization on each video to link all “tubes”.

Unsupervised methods are more related to segmentation and attempt to find relevant moving parts in a sequence (Ma et al., 2013; Yu and Yuan, 2015). Ma et al.localize performers inferring the bounding box location from the set of HSTS (Ma et al., 2013). Yu et al. formulate the action proposal localization as a maximum set coverage problem. They use greedy search to select action proposal maximizing an actionness score computed along a spatio-temporal path (Yu and Yuan, 2015). Since they are unsupervised methods, some classification algorithm has to be used in conjunction in order to localize actions of a given class otherwise such algorithms must be regarded as a mean of salient objects extraction.

The remainder of the paper is organized as follows: in Section 2 we formally define action recognition as a video matching problem introducing our framework; in Section 3 we present our video representation algorithm and in Section 4 we define our cluster set kernel; Section 5 extends our action recognition algorithm enabling action localization; experimental results are presented in Section 6 and conclusions are drawn in Section 7.

2. Problem formulation

Automatic annotation of video content requires the definition of a representation for videos and a similarity function to compare such representations. Classifiers, like kernelized SVMs, can be learnt straightforward once these two components are defined. Recent contributions in action recognition have shown that methods based on local trajectory aligned descriptors are the best performing (Wang et al., 2013; 2015a). These methods represent videos with a set of local descriptors.

Our video matching problem can then be framed as a the comparison between two sets of possibly different cardinalities. Sum Match Kernels (SMK) have been proposed in the past as a solution to preserve local features quality and avoiding codebook quantization (Wallraven et al., 2003). Global Pooling, aggregating all features in a single signature, is often used instead because of its simplicity and computational advantage. Our method, as also shown in Fig. 3, is a middle ground between these two approaches.

Let $f(x, y)$ be a kernel measuring the similarity of two local features x and y belonging to sets $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively. A sum match kernel is defined as

$$K(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{x \in \boldsymbol{X}} \sum_{y \in \boldsymbol{Y}} f(x, y) \tag{1}$$

An interesting property of SMKs is the fact that they allow to compute correspondences between any visual structure captured by the local features. However, they have a major drawback, since in their all-vs-all approach, if only a small set of features between the two sets has a high similarity, this signal may be cancelled by the many low-scoring feature correspondences. Sum-max kernel and power match kernel have been proposed to deal with this issue (Seidenari et al., 2014; Wallraven et al., 2003). Moreover the exhaustive matching procedure that they require has a very high computational cost, which is aggravated in our case by the large amount of local features detected in videos.

Pooling based approaches can be interpreted as efficient approximations of SMKs. First they code local features as high dimensional embeddings and then these embeddings are pooled together in a usually high dimensional signature. A kernel obtained by the comparison of these signatures is then equivalent to the SMK computed on the encodings. Defining an encoding function $\psi(x)$, that embeds a feature x in a higher dimensional space, if $f(\cdot, \cdot)$ is an

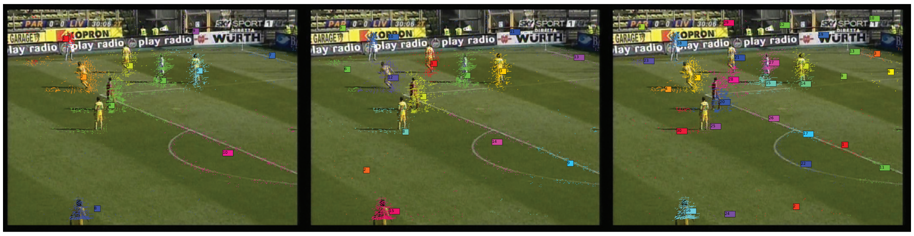
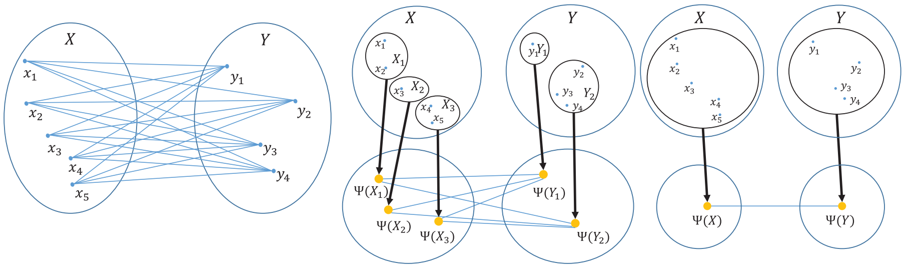


Fig. 2. Automatically clustered trajectories on soccer dataset (10, 15 and 30 clusters). Several clusters gather features of a single player. Noisy clusters often capture textured regions of the background.



(a) Match Kernel (b) Cluster Set Kernel (c) Global Pooling

Fig. 3. We show three possible set matching models. Blue lines indicate correspondence operations, blue points indicate local features and yellow bold points indicate pooled features. Match Kernels (a) perform an exhaustive comparison, while Global Pooling (c) performs a single comparison. Our Cluster Set Kernel, compares spatio-temporally consistent subsets of features reducing the amount of comparisons and increasing the discriminativity. Note that for video features, $|\boldsymbol{X}| \simeq 10^5$ while $|\mathcal{P}(\boldsymbol{X})| \simeq 10$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

additive kernel, it can be shown that

$$K(\boldsymbol{X},\boldsymbol{Y}) = \sum_{x\in\boldsymbol{X}}\sum_{y\in\boldsymbol{Y}} f(\psi(x),\psi(y)) = f\left(\sum_{x\in\boldsymbol{X}}\psi(x), \sum_{x\in\boldsymbol{X}}\psi(y)\right) \qquad (2)$$

Defining

$$\Psi(\boldsymbol{X}) = \sum_{x\in\boldsymbol{X}}\psi(x) \qquad (3)$$

Eq. (2) can be simply rewritten as

$$K(\boldsymbol{X},\boldsymbol{Y}) = f(\Psi(\boldsymbol{X}), \Psi(\boldsymbol{Y})) \qquad (4)$$

Global Pooling approaches are interesting since they formally approximate match kernels and are extremely more efficient to compute (Bo and Sminchisescu, 2009; Sánchez and Perronnin, 2011). Although, they do not allow to compare subsets of features. We believe that action recognition can be improved if spatio-temporally consistent subsets of video features can be put in direct correspondence without a global pooling. This idea represents a trade-off between the exhaustive comparison of SMK and the single signature approach of global pooling.

We therefore propose to apply pooling to subsets of local features, that should be grouped according to their spatio-temporal properties. Each subset of features is represented through global pooling. Finally, the similarity of two videos is obtained with an exhaustive comparison between all pairs of signatures. This approach has two main motivations, first of all it creates a more discriminative kernel function, second it avoids the cost of an exhaustive comparison of all pairs of features.

Our approach can be modelled as

$$K(\boldsymbol{X},\boldsymbol{Y}) = \sum_{\boldsymbol{X}_i\in\mathcal{P}(\boldsymbol{X})}\ \sum_{\boldsymbol{Y}_i\in\mathcal{P}(\boldsymbol{Y})} f(\Psi(\boldsymbol{X}_i), \Psi(\boldsymbol{Y}_i)). \qquad (5)$$

where $\mathcal{P}(\boldsymbol{X})$ is a partition of set $\boldsymbol{X}$ such that

$$\bigcup_{X_i\in\mathcal{P}(\boldsymbol{X})} \boldsymbol{X}_i = \boldsymbol{X} \text{ and } \bigcap_{X_i\in\mathcal{P}(\boldsymbol{X})} \boldsymbol{X}_i = \emptyset. \qquad (6)$$

3. Video representation

Our video representation is designed to capture the spatio-temporal structure of the video. In team sports, activities are often defined just by a subset of the players. Ideally mapping visual features to players or other relevant elements (e.g. the ball, the referee etc.), allows to obtain a detailed representation of the scene, however player tracking and detection is an extremely challenging task that is prone to failure. Failing to track or detect players or other relevant entities breaks the recognition pipeline leading to inconsistent results.

Our method is more robust and consists of two main steps: a feature partitioning step and a matching step. First we group trajectories in an unsupervised manner with an efficient method allowing to deal with the several thousands of features extracted per frame, and then we use our cluster match kernel that allows to make correspondences among the grouped trajectories. We learn one-vs-rest classifiers using SVM and this kernel. Clusters can be used to generate action proposals that allow to learn per-frame action localisers. A diagram showing the data flow of our method is shown in Fig. 4.

3.1. Trajectory clustering

We want to identify discriminative spatio-temporal structures, in order to better distinguish the actions performed in the videos. To achieve this goal, we employ a spectral clustering technique on the extracted trajectory features. Due to the large amount of features extracted by the IDT algorithm, we choose an optimised spectral clustering algorithm, namely Landmark Based Spectral Clustering (LSC) (Chen and Cai, 2011). In Fig. 2 the output of the feature grouping algorithm shows that clusters roughly localize players.

Spectral clustering is a relaxation of Normalised Cut algorithm that tries to exploit the connectivity of data. Spectral clustering exploits the eigenvalues of the Laplacian to obtain a better representation that allows to easily separate clusters using K-Means.

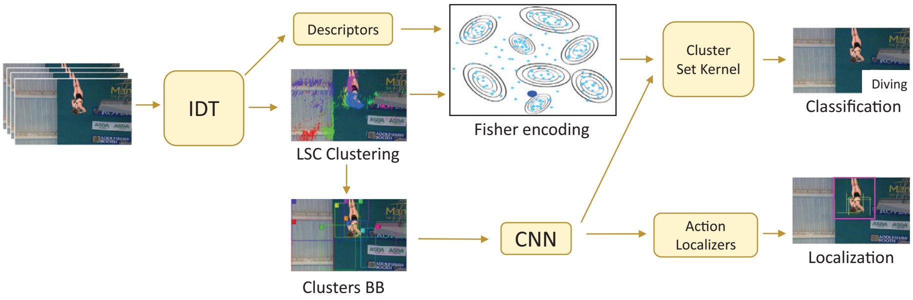


Fig. 4. The workflow of our action recognition and localization framework. Trajectories are extracted from every clip using IDT. Each trajectory is associated with local feature descriptors (Descs). Trajectories are grouped using Landmark Based Spectral Clustering (LSC). Each cluster descriptor set is encoded as a Fisher Vector. Clusters are used to generate per-frame action proposals. Activations from a Convolutional Neural Network (CNN) are used on action proposals to train per-frame action localisers. We use max-pooled features of the second-last fully connected layer, from clusters together with Fisher Vectors in our Cluster Set Kernel.

The main problem with big input data is computing the graph Laplacian and its factorization. Indeed the computational complexity for an eigenvalue problem is $O(n^3)$. To reduce this cost, the approach of LSC is to first project data in a smaller space and then apply the eigenvalue decomposition on such reduced size problem.

Let $\mathbf{L} = [\boldsymbol{l}_1, \ldots, \boldsymbol{l}_n] \in \mathbb{R}^{m \times n}$ be the data matrix. First we sample the input data to obtain two matrices: $\mathbf{U} \in \mathbb{R}^{m \times p}$, the landmarks matrix, and $\mathbf{Z} \in \mathbb{R}^{p \times n}$, the data projected in a smaller space of size $p \ll n$. By this way we can approximate $\mathbf{L} \approx \mathbf{UZ}$, thus making Laplacian and eigenvectors computation more lightweight. Preselection of landmarks is performed using K-Means, however random point selection is also feasible. These samples are the basis vectors used to represent the input data in a reduced space.

Given the samples and matrix $\mathbf{U}$, the elements of the sparse representation matrix $\mathbf{Z}$ can be calculated efficiently as

$$z_{ji} = \frac{K_h(\boldsymbol{l}_i, \boldsymbol{u}_j)}{\sum_{j \in U} K_h(\boldsymbol{l}_i, \boldsymbol{u}_j)} \tag{7}$$

Where $K_h(\cdot)$ is a kernel function, in our case the Gaussian kernel $K_h(\boldsymbol{l}_i, \boldsymbol{u}_j) = \exp(-\frac{||\boldsymbol{l}_i - \boldsymbol{u}_j||^2}{2h})$. We now can compute the eigenvalues and eigenvectors of $\mathbf{ZZ}^T$, choosing the first k and applying K-Means to obtain the clusters. The clustering pseudocode is outlined in Algorithm 1.

Algorithm 1: `LSCClustering`.

```
Data: $n$ data points $\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_n \in \mathbb{R}^m$, Cluster number $k$
Result: Indices of $k$ Clusters
1 Choose $p$ landmarks using a K-Means pass with few iterations
2 Compute matrix $\mathbf{Z} \in \mathbb{R}^{p \times n}$ as shown in Equation (7)
3 Compute the first $k$ eigenvectors of $\mathbf{ZZ}^T$,
  $\mathbf{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k] \in \mathbb{R}^{k \times p}$ and eigenvalues $(\sigma_1, \ldots, \sigma_k)$
4 Compute $\mathbf{B}^T = \Sigma^{-1}\mathbf{V}^T\mathbf{Z}$, where $\Sigma = diag(\sigma_1, \ldots, \sigma_k)$
5 Apply K-Means to $\mathbf{B}$ rows to obtain the indices vector $\boldsymbol{I}$ of the $k$ clusters for the $n$ input observations
6 return $(\boldsymbol{I})$
```

Considering step 3 of Algorithm 1, being $\mathbf{ZZ}^T \in \mathbb{R}^{p \times p}$, the eigenvalue decomposition computational cost is then $O(p^3)$ while the cost of step 4 is $O(p^2 n)$, considering the fact that in our case $n \simeq 10^4$ while $p \simeq 10^3$, there are at least three orders of magnitude of difference in practice.

3.2. Cluster representation

We represent local features with HoG, HoF, MBH and trajectory descriptors concatenating the normalised spatio-temporal coordinates to the local descriptors. HoG, HoF and MBH are respectively histograms of gradients, optical flow and motion boundaries, i.e. derivatives of optical flow, while trajectory descriptors are the concatenation of trajectory coordinates normalised by the length of the trajectory (Wang and Schmid, 2013). Each cluster is represented with a Fisher Vector encoding of the local descriptors that have been assigned to it.

Our approach is agnostic regarding the local descriptors and the feature aggregation method. We decided to build on IDT and Fisher Vectors which are shown to perform consistently on several datasets since our focus is on trajectory grouping and matching.

We apply PCA retaining the first 80 components of all histogram features and 20 of the trajectory descriptors; we concatenate the normalised spatio-temporal coordinate of each trajectory centre to the PCA-compressed local feature in order to retain information about the global location. We learn a codebook of 256 Gaussians using GMM. PCA and GMM codebook are learned on a random sample of 200K training features. Fisher Vectors are calculated using the Improved algorithm which applies both L_2-normalization and power normalization (Perronnin et al., 2010).

Given a Gaussian Mixture Model with parameters $\boldsymbol{\mu}_n$, $\boldsymbol{\sigma}_n$, $\boldsymbol{\omega}_n$ and given soft-assignments $\gamma_m^{(n)}$ for each of the M augmented local feature $\boldsymbol{x}_m \in \boldsymbol{X}$, the Fisher Vector is computed concatenating the likelihood gradients:

$$\Psi(\boldsymbol{X}) = \left[\mathcal{G}_n^{\mu}(\boldsymbol{X}) \; \mathcal{G}_n^{\sigma}(\boldsymbol{X})\right] \tag{8}$$

where

$$\mathcal{G}_n^{\mu}(\boldsymbol{X}) = \frac{1}{\sqrt{\boldsymbol{\omega}_n}} \sum_{m=1}^{M} \gamma_m^{(n)} \left(\frac{\boldsymbol{x}_m - \boldsymbol{\mu}_n}{\boldsymbol{\sigma}_n^2} \right), \tag{9}$$

$$\mathcal{G}_n^{\sigma}(\boldsymbol{X}) = \frac{1}{\sqrt{2\boldsymbol{\omega}_n}} \sum_{m=1}^{M} \gamma_m^{(n)} \left(\frac{(\boldsymbol{x}_m - \mu_n)^2}{\boldsymbol{\sigma}_n^2} - 1 \right), \tag{10}$$

and

$$\gamma_m^{(n)} = \frac{\boldsymbol{\omega}_n p_n(\boldsymbol{x}_m)}{\sum_{j=1}^{D} \boldsymbol{\omega}_j p_j(\boldsymbol{x}_m)}, \tag{11}$$

4. Video matching

Clustering the set of augmented local features $\boldsymbol{X}$ extracted from a video, according to Algorithm 1, yields a partition $\boldsymbol{P}(\boldsymbol{X})$.

Based on the formulation introducted in Section 2 we define a kernel inspired by Match Kernels (Wallraven et al., 2003) that exploits trajectory grouping to reduce the matching complexity and to compute correspondences among coherent subset of video features.

Please cite this article as: F. Turchini et al., Understanding and localizing activities from correspondences of clustered trajectories, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2016.11.007

4.1. Cluster set kernel

Given a pair of videos and their respective feature sets $\boldsymbol{X}$ and $\boldsymbol{Y}$, after the clustering step we compute our cluster set kernel as follows

$$K(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{|\mathcal{P}(\boldsymbol{X})|}\sum_{X_i\in\mathcal{P}(\boldsymbol{X})}\max_j \Psi(\boldsymbol{X}_i)^T\Psi(\boldsymbol{Y}_j) + \frac{1}{|\mathcal{P}(\boldsymbol{Y})|}\sum_{Y_j\in\mathcal{P}(\boldsymbol{Y})}\max_i \Psi(\boldsymbol{X}_i)^T\Psi(\boldsymbol{Y}_j) \tag{12}$$

where $\mathcal{P}(\boldsymbol{X})$ is a partition of $\boldsymbol{X}$ defined as in Eq. (6).

In this way, we obtain a symmetric kernel matrix. Our kernel takes into account the similarity scores both of $\boldsymbol{Y}$ respect to $\boldsymbol{X}$ and of $\boldsymbol{X}$ respect to $\boldsymbol{Y}$. If two videos are similar, we should obtain high scores from both operations, and thus a high combined score. Even though our kernel can not formally satisfy the Mercer property it has been shown that this is not a strict requirement for an SVM classifier to learn an accurate solution. In practice the kernel matrices we have computed were always positive definite so far.

Given different groupings $\mathcal{P}(\boldsymbol{X}_n^f)$ for each feature f and the respective kernels $K(\mathcal{P}(\boldsymbol{X}_n^f),\mathcal{P}(\boldsymbol{Y}_n^f))$ our final kernel can be computed as

$$K(\boldsymbol{X},\boldsymbol{Y}) = \sum_f\sum_n K(\mathcal{P}(\boldsymbol{Z}_n^f),\mathcal{P}(\boldsymbol{Y}_n^f)) \tag{13}$$

thus integrating different local representation and spatio-temporal structures. The formulation in Eq. (13) allows to fuse multiple trajectory partitionings, with different cluster cardinalities, and local feature descriptors.

We normalize the kernel obtained from Eq. (13) using the following

$$\hat{K} = DKD^T \tag{14}$$

where D is a diagonal matrix such that $D_{ii} = 1/\sqrt{K_{ii}}$.

4.2. Quantized kernel

Our approach requires the storage and processing of a large amount of high dimensional feature vectors. This problem is also worsened by the fact that we employ multiple features.

Jegou et al. (2011) proposed to use Product Quantization (PQ) to reduce the dimensionality of feature vectors. Perronin et al. applied the same idea to Fisher Vectors (Sánchez and Perronnin, 2011) and the idea of reducing data storage and access cost with this technique is considered a good practice for large scale learning (Akata et al., 2014). Differently from Vector Quantization (VQ), which acts mapping a whole high dimensional vector onto a single representative, PQ splits signatures in several blocks learning a quantizer for each sub-vector.

There are two sources of spatial complexity burden in our framework: local features extracted by IDT, and high dimensional signatures representing clusters. We only quantize the final signature of a cluster X, $\Psi(X)$, while local features, once encoded as described in Section 3, can be discarded. Moreover, quantizing local features leads to degraded performance, affecting the quality of trajectory clustering and of the overall representation.

We learn a product quantizer $\mathcal{Q}$ using K-Means which outputs a set of M codebooks. The quantizer $\mathcal{Q}$ is a set of quantizing functions $q_i(x)$ mapping a feature x onto a codeword $c_i^1 \dots c_i^b$. A Fisher Vector $\Psi(X)$ will be represented by an ordered set of indices $\mathcal{Q}(\Psi(X)) = \{q_i(\Psi_i(X))\dots q_M(\Psi_M(X))\}$ where $q_i(\ \cdot\ )$ is a quantizer function learned for the i-th vector block $\Psi_i(X)$. Instead of reconstructing the features as in Sánchez and Perronnin (2011),Akata et al. (2014) we directly precompute dot products among all codewords of each quantizer thus defining $\mathbf{P}^i = [c_i^1 \dots c_i^b]^T[c_i^1 \dots c_i^b]$.

The approximated dot product for two features $\Psi(X)$, $\Psi(Y)$ becomes:

$$\Psi(X)^T\Psi(Y) \approxeq \sum_{m=1}^{M}\mathbf{P}^m_{q_m(\Psi(X))q_m(\Psi(Y))} \tag{15}$$

where $q_m(\Psi(X)) = q_m(\Psi_m(X))$ to ease the notation. Plugging Eq. (15) into Eq. 13 we obtain our quantized kernel:

$$K(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{|\mathcal{P}(\boldsymbol{X})|}\sum_{X_i\in\mathcal{P}(\boldsymbol{X})}\max_j\sum_{m=1}^{M}\mathbf{P}^m_{q_m(\Psi(X_i))q_m(\Psi(Y_j))} + \frac{1}{|\mathcal{P}(\boldsymbol{Y})|}\sum_{Y_j\in\mathcal{P}(\boldsymbol{Y})}\max_i\sum_{m=1}^{M}\mathbf{P}^m_{q_m(\Psi(X_i))q_m(\Psi(Y_j))} \tag{16}$$

Different compression rates can be obtained varying the two parameters G and b, which are respectively block size of $\Psi(X)$ used to dived it in M blocks and the amount of codewords used for each block.

4.3. Complexity analysis

Following the model presented in Section 2, we conduct a more detailed analysis of the computational cost of our method. Let us first consider a Global Pooling approach, such as Fisher Vectors. To simplify the analysis we split the cost in *encoding* and *matching*. Encoding refers to the phase in which local features are coded into a higher dimensional representation and then pooled to form a single high dimensional signature. Matching refers to the step required to compute the similarity between two of such signatures.

Encoding L local features with a dimensionality of D, for a dictionary of K Gaussians requires $O(2KDL)$, while matching is $O(2KD)$. The encoding step defined by Eqs. (9) and (10) requires the calculation of L differences, for D dimensional feature, for each of the K Gaussian n. Matching is performed with a scalar product between two $2KD$-dimensional feature vectors.

Let us now consider our Cluster Set Kernel approach. In our case the encoding step must be performed for each cluster of each video, however each cluster contains just a subset of the whole video feature set. Considering Eq. (6), we can conclude that the encoding step has the same cost of a single Global Pooling, i.e. $O(2KDL)$, since the amount of local features to be encoded (L) is the same in both approaches. Regarding the matching step, we have a higher cost, since in Eq. (12) we compute scalar product among all $2KD$-dimensional cluster encodings. Considering a partitioning of features in N clusters, for both videos, the matching cost is $O(2KDN^2)$.

In case the Match Kernel approach is used, assuming that videos have roughly the same amount of local features L, the complexity is $O(DL^2)$. If we compare the cost of Global Pooling and our Cluster Set Kernel, it is evident that both quantities are dominated by the encoding time. For Global Pooling $O(L2KD) > O(2KD)$ and for our approach a similar relation holds, $O(L2KD) > O(N^2 2KD)$, since practically $L \gg N^2$. Finally if we consider the cost of a Match Kernel compared to our approach, the ratio of the two costs is $\frac{L}{2K}$, and since $L \simeq 10^5$ and $K \simeq 10^2$, the Match Kernel has a cost of three orders of magnitude higher with respect to our approach.

5. Action localization

Our clusters are able to reveal the structure of a video decomposing the feature set into spatio-temporally consistent subsets. The kernel defined in Eq. (13) allows a partial matching of the feature set to be performed, however it still treats the video globally, hence not allowing a proper action localization. In this section we show how our clusters can be exploited to generate a set of action proposals in each frame and learn action localizers. Moreover, proposal regions can be used to sample new features computed on single frames thus enriching the cluster representation beyond the IDT local descriptors.

5.1. Action proposals

Given a frame from a sequence, classified as action $\boldsymbol{y}$, we define an action proposal as a sub-region of the frame likely to contain, partially or completely, the person performing such action. Trajectory clusters are spatio-temporal entities with an extent that may or may not encompass the whole video. For each frame we are able to extract a set of bounding boxes from each live cluster. We consider a cluster $\boldsymbol{X}_i$ alive at frame t if there exists at least one trajectory, belonging to $\boldsymbol{X}_i$, in that frame.

Consider a cluster $\boldsymbol{X}_i$ and a frame at time t, we obtain an action proposal, defined by a box $B_i(t) := [x, y, h, w]$ as in the following:

$$x = \min_{\boldsymbol{X_i}} x_{traj}(t), \tag{17}$$

$$y = \min_{\boldsymbol{X_i}} y_{traj}(t), \tag{18}$$

$$w = \max_{\boldsymbol{X_i}} x_{traj}(t) - x, \tag{19}$$

$$h = \max_{\boldsymbol{X_i}} y_{traj}(t) - y, \tag{20}$$

where $x_{traj}(t)$, $y_{traj}(t)$ are the trajectory coordinates at frame t. Eqs. (17)–(20), formally define the extent of $B_i(t)$, which is simply the rectangular region containing all trajectories from cluster $\boldsymbol{X}_i$ at frame t.

We represent each rectangular region using the activation from the second-last fully connected layer of a convolutional neural network (Girshick et al., 2014). We use the features learned from the pre-trained VGG-16 network (Chatfield et al., 2014) to represent each bounding box after warping it to a 224 × 224 square region. Transfer learning is performed simply by training a linear classifier over the CNN codes computed from VGG-16, as detailed in the following Section. This approach has been shown to be successful for a variety of tasks including action recognition (Ravanbakhsh et al., 2015; Sharif Razavian et al., 2014).

In the following, we discuss how to incorporate the frame-wise knowledge that we are able to extract through proposals. We exploit action proposals in a twofold manner. First we use the frame-wise local features to learn action localizers, then we devise a method to represent clusters with these features and incorporate the representation into Eq. (13) to improve classification.

5.2. Detection

We now detail a strategy to form a training set for frame-wise action localizers, based on our action proposals. We consider a proposal “positive” if it has an Intersection over Union (IoU) with the ground truth higher than 0.3. Object detections are usually considered correct if IoU exceeds 0.5 (Everingham et al., 2010), however we use this more permissive value in order to obtain a sufficient amount of positive samples. Given a category, we collect a set of positive and negative box samples, from each training video. For the positive set we avoid boxes from clusters which are composed by more than the 80% of negative samples. We perform this pre-filtering to avoid spurious overlapping boxes from mostly negative clusters contaminating the positive training set. The full training set is then obtained by merging all samples from all training videos.

We learn a binary linear SVM for every class, over positive and negative samples from all training videos, obtaining our action detectors. At test time, for each frame, we retain only bounding boxes with a positive score and to obtain a single bounding box we perform the following procedure. We build an energy map accumulating the scores of positive boxes with a procedure similar to Karaman et al. (2014). Energy maps are min-max normalized and thresholded with a value of 0.5. The final box, predicting the action location, is the one enclosing the largest peak.

This procedure is different from Non-Maximal Suppression (NMS) which is the algorithm of choice when performing detection and there is the need of simplifying a set of redundant boxes. Indeed NMS can only be applied when samples are clustered with a high overlap. Our proposals are generated directly from clusters which are learned unsupervisedly with the objective of partitioning video trajectories into separated and consistent sets. Our proposals usually focus on the details of actions, like a moving arm or the upper body bending. Therefore our procedure allows to compute a higher quality bounding box for the action to be predicted and to reduce the influence of outliers.

5.3. Improving classification

Our frame-wise proposals, computed using Eqs. (17)–(20), allow to associate to each cluster a set of CNN activations. We exploit this powerful feature to improve classification. We obtain a single feature vector for each cluster by max-pooling over all the activations. We can now compute the kernel defined in Eq. (12) and combine it with kernels computed from IDT features using Eq. (13). As for Fisher Vectors we use a dot product to compare max-pooled activations.

6. Experiments

To test our framework, we performed experiments on a generic sport dataset (UCF Sports), and two specific sport datasets, namely MICC-SOCACT4 and Volleyball Activity Dataset 2014. We also tested our method on HighFive and Hollywood2 which are two popular and challenging datasets of generic actions to show that our method is extremely general and is applicable not just to sport activities but also to generic action recognition. We also evaluated the trade-off between accuracy and compression rate obtained by our quantized kernel and the localization accuracy of our cluster based proposals.

6.1. Datasets

UCF Sports. UCF Sports is comprised of 10 actions selected from various sports and recorded from TV broadcast (Diving, Golf Swing, Kicking, Lifting, Riding Horse, Running, Skateboarding, Swing-Bench, Swing-Side, Walking). There are 150 scenes at 720x480 resolution. For action recognition, we use the Leave-One-Out (LOO) cross-validation scheme. For the localization task we use the same setting of Lan et al. (2011).

MICC-SOCACT4. This dataset is composed by 100 MPEG-2 videos at PAL resolution (720x576). These videos represent 4 soccer actions: “Goal Kick”, “Throw In”, “Placed Kick”, “Shot on Goal” and were recorded from 5 different matches of the Italian “Serie A”. We picked a match as the test set and the other 4 as the training set, performing a 5-fold cross-validation.

Volleyball Activity Dataset 2014. This dataset is composed by 6 full volleyball matches of the Austrian Volley League originally recorded in full HD resolution. They were annotated with 7 classes, 5 specific volley classes (“Serve”, “Reception”, “Setting”, “Attack”,

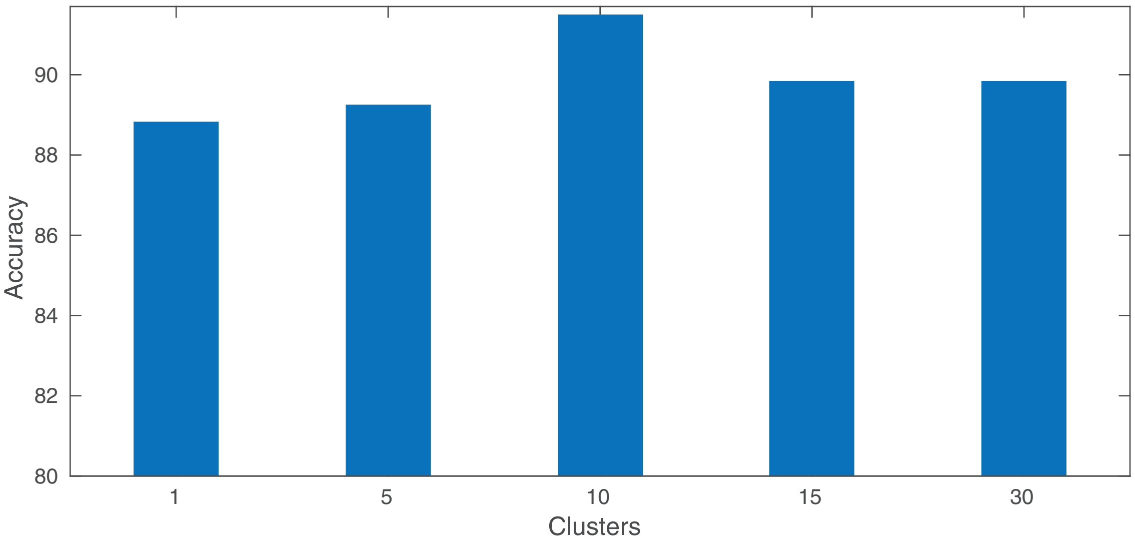


Fig. 5. Accuracy values varying number of clusters for the MICC-SOCACT4 dataset.

"Block"), and 2 more general classes ("Stand", "Defense/Move"). We take in exam the tracklets, which represent the continuous player activities lasting about 1–2 seconds. We cut the original videos according to the tracklets, obtaining about 900 videos, that we used as the actual classification dataset. The cut script provided by the authors crops the area around annotated players, while we take the entire frame in the same time window, reducing its resolution to 640x360 and adding additional 15 frames at the beginning and 15 at the end of the tracklet. Data was partitioned in 50% for training and 50% for testing, according to Waltner et al. (2014). Results are reported for two experimental settings: recognition of all the seven activities and recognition of only the five, volleyball specific, activities.

Hollywood 2. The Hollywood 2 dataset is composed of 1707 clips split in training and test. We only use the "clean" annotated set. The dataset has 12 classes of human actions and is composed of roughly 20 hours of video in total. It is considered one of the most comprehensive and challenging benchmark for human action recognition in realistic settings. The dataset is composed of video clips from 69 movies extracted from DVDs. To avoid any bias the training clips are sampled from a different set of movies with respect to the test clips.

HighFive. The HighFive dataset is focused on the recognition of human interactions. It is collected from TV series and contains 4 classes and a set of negative examples. The dataset is composed by 300 videos and a training/testing split is proposed by Patron-Perez et al. (2012). Results are reported computing mean average precision on all classes except the negative one.

6.2. Action recognition

We evaluate the accuracy in activity recognition of our proposed method on several public datasets. We argue that our method is not tailored to specific sports nor require adaptation to cope with viewpoints that are characteristic of certain sport broadcast. For this reason we also test our method on generic action recognition datasets, showing state of the art results.

In all experiments we extracted improved dense trajectories descriptors with the default parameters without using human detection to filter trajectories. On smaller datasets such as MICC-SOCACT and UCF Sports, we also extracted descriptors on flipped versions of videos.

Throughout Section 6, we use three variations of our method. We set as baseline a standard Fisher Vector pipeline with linear SVM classifiers. Considering Eq. (12) this baseline is equivalent to applying our framework with a single cluster per video sample grouping all features. We refer to this variation as "Fisher Vector baseline". To evaluate our clustering based representation, we use a simpler version of our method, based on Eq. (12), using a single cluster cardinality, e.g., 10. We refer to this simplified version of our approach as "Our Clustering". Finally, the complete approach considering multiple cluster cardinalities, implemented in Eq. (13), is referred as "Our Fusion".

We evaluate first how parameters of trajectory clustering affect recognition accuracy. In Figs. 5 and 6 we report how the classification accuracy varies depending on the number of clusters used. In this experiments we could note how the classification accuracy does not depend strongly on the amount of clusters used per video, anyways the best performance on both dataset is obtained using 10 clusters. Kernels computed with different cluster cardinalities lead to similar accuracy but likely capture different details of the imagery. Indeed our fusion approach, as discussed in the following sections, consistently leads to superior performance on all datasets.

We then evaluate how the accuracy of our method is affected by the number of Gaussians employed in the dictionary construction. As shown in Table 1, performance for codebooks larger than 64 Gaussians saturates.

6.2.1. Experiments on sports datasets

We first show a comparison with the state-of-the-art on UCF Sports Actions, which highlights the good behaviour of our method with respect to other known approaches. We show competitive results, beated only by Ravanbakhsh et al. (2015), using CNN features from image pretrained networks, over snippets of frames.

We outperform (Karaman et al., 2014), without making use of pooling maps to weight high saliency areas of the scenes (Karaman et al., 2014). UCF Sports actions are performed by individual athletes, so the clustering step is able to put in evidence the salient subset of trajectories without additional external information. Note that the method proposed by Vrigkas et al. (2014), requires bounding box for each action in each frame, while we do not.

We report performance in the recognition of specific sport activities in Soccer and Volleyball. First we report results of our method on the smaller MICC-SOCACT4 dataset. Soccer actions are

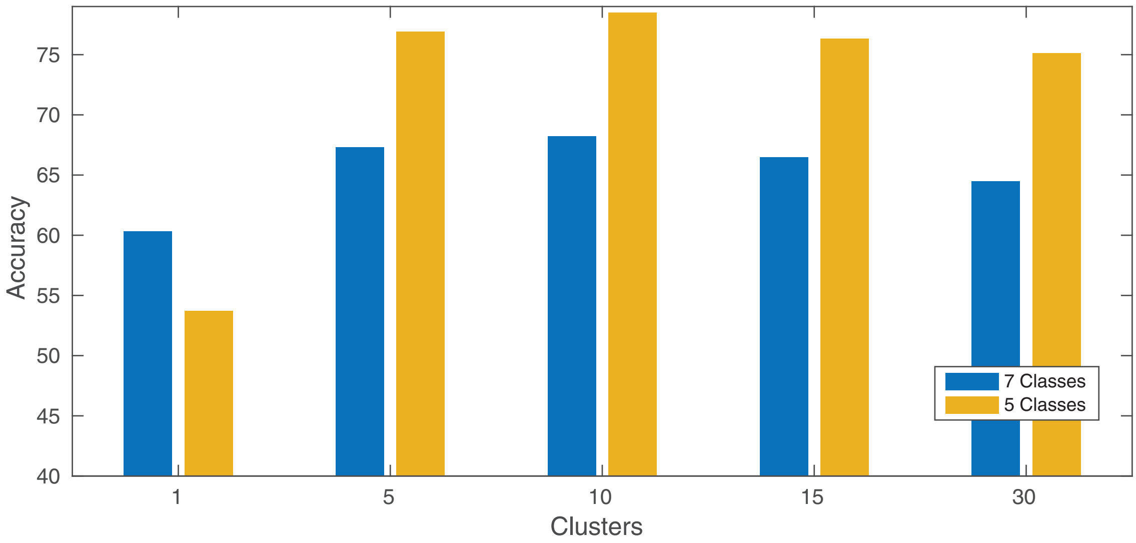


Fig. 6. Accuracy values varying number of clusters for the Volleyball Actions 2014 dataset.

Table 1
Mean Average Precision on HighFive dataset varying the size of Mixture of Gaussians codebook. Above 128 Gaussians it can be observed that mAP saturates.

# GMM	Fisher vector	Our clustering	Our fusion
64	66.4	75.7	75.8
128	68.3	77.0	77.3
256	68.5	77.1	77.6
512	69.4	77.3	77.6

Table 4
Mean per class accuracy results on the Volley-Ball Activity dataset compared with (Waltner et al., 2014).

Method	Acc. 7 Cl.	Acc. 5 Cl.
Our fusion	**91.2**	**94.1**
Our clustering	68.2	78.5
Fisher vector baseline	60.3	53.7
Waltner et al. (2014)	77.5	90.2

Table 2
Comparison with the state of the art on the UCF Sports dataset. Results are reported as mean per-class accuracy over the 10 classes.

Method	Accuracy
Our fusion	**92.9**
Our clustering	90.5
Fisher vector baseline	88.6
Ravanbakhsh et al. (2015)	**97.8**
Vrigkas et al. (2014)	95.1
Karaman et al. (2014)	90.4
Wang et al. (2013)	89.1
Lan et al. (2011)	83.7
Kovashka and Grauman (2010)	87.3
Kläser (2010)	86.7
Wang et al. (2009)	85.6
Yeffet and Wolf (2009)	79.3
Rodriguez et al. (2008)	69.2

Table 3
Mean per class accuracy of our method compared with (Ballan et al., 2010) on the MICC-SOCACT4 dataset.

Method	Accuracy
Our fusion	**92.5**
Our clustering	91.5
Fisher vector baseline	88.8
String Kernel+SVM (Ballan et al., 2010)	73.0
NN+NWD (Ballan et al., 2010)	54.0

often defined by collective behaviours. On this dataset, our Fisher Vector baseline already improves over (Ballan et al., 2010) by a large margin as is shown in Table 3. Nevertheless our correspondence kernel can boost the accuracy further obtaining 92.5%, especially raising the accuracy on “Goal Kick” and “Placed Kick” as can be seen from the confusion matrices in Fig. 8; in both these actions there is a single player performing a discriminative motion: kicking the ball from a fixed position, while other players are less involved in the action. For this reason our clustering can isolate these actions and better match the respective spatio-temporal structures.

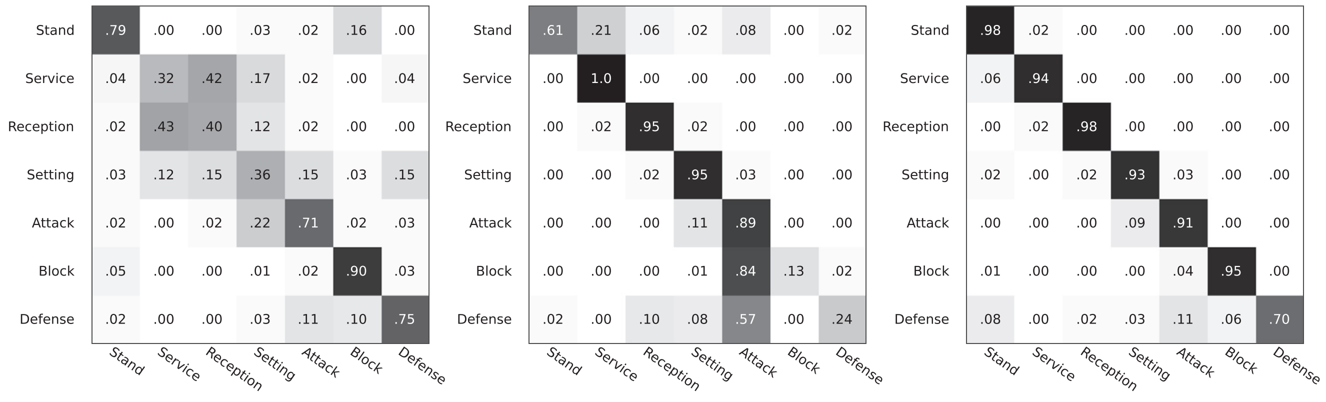
In Table 4 we report a comparison of our method with previous work and our baselines on the Volleyball Activity Dataset. It can be seen that our baseline with a single Fisher Vector per video performs worse than (Waltner et al., 2014). Our clustering baseline improves over the FV baseline by 8% (and by 25% on 5 classes). Some player activities are better recognized in isolation as can be seen in the confusion matrix while other are better recognized exploiting context. From Fig. 7 it is clear that collective activities as “Block” and “Defence” are better captured by a global representation (FV), while individual actions like “Attack” and “Service” are better recognized by our correspondence kernel.

The fusion approach implemented by Eq. (13) is able to obtain accurate results in both setups outperforming the state of the art by more than 14% (and by 4% on 5 classes).

The classification task noticeably benefits from the clustering step, especially on the volleyball dataset. However, it appears clearly, looking at the confusion matrices in Fig. 7, that results are complementary. “Stand”,“Block” and “Defence” need some additional contextual information to be recognized, while clustering focuses on local information located in cluster areas, which is better for the other classes. Notice how the fusion allows to distinguish between Block and Attack actions, which are almost totally confused by the clustering method.

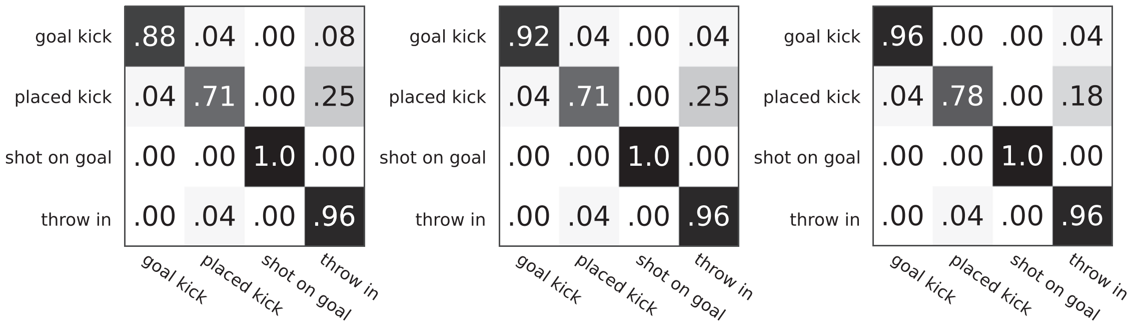
On soccer videos classification accuracy took just a small advantage from the fusion, compared to clustered Fisher encoding by itself. We can hypothesize that soccer scenes do not benefit from global contextual information because of their structure and dy-

Please cite this article as: F. Turchini et al., Understanding and localizing activities from correspondences of clustered trajectories, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2016.11.007



(a) Baseline (b) Our Clustering (c) Our Fusion

Fig. 7. Confusion matrices for volleyball.



(a) Baseline (b) Our Clustering (c) Our Fusion

Fig. 8. Confusion matrices for soccer. Our method improves on "Goal Kick" and "Placed Kick" actions.

namic characteristics, with very high camera motion that is not fully compensated by improved trajectories features, while volley sequences need to be analysed both globally and in specific areas to locate the distinctive elements, such as the players arrangement.

6.2.2. Experiments on generic actions

We show results on two popular and challenging generic action recognition datasets: HighFive (Patron-Perez et al., 2012) and Hollywood2 (Laptev et al., 2008). We report mean Average Precision (mAP) on the full test set, following the standard protocol for Hollywood2. Considering the multi-label nature of Hollywood2, we could not compute Confusion Matrices using the whole dataset. For the sole estimation of confusion matrices, videos with more than one action, ∼ 10% of the dataset, were removed. This operation is necessary in order to correctly estimate the aforementioned matrices.

As shown in Table 5, on the Hollywood2 dataset our baseline is comparable to the result proposed by Wang et al. (2015a) with a 0.1% difference. Clustering alone has a similar performance and the fusion is above the state of the art obtaining 70.7% of mAP. On this challenging dataset our clustering approach offers a high performing, complementary representation, to the global approach. Hollywood2 contains challenging untrimmed sequences where multiple actions can be performed at once (e.g. Hugging and Kissing). We believe that matching local patterns improves the recognition given the complex structure of video sequences in this dataset. Confusion matrices on Hollywood2, reported in Fig. 10, show that our Fusion method reduce the misclassifications.

Table 5
Comparison with the state of the art on the Hollywood2 dataset. Results are reported as mean average precision.

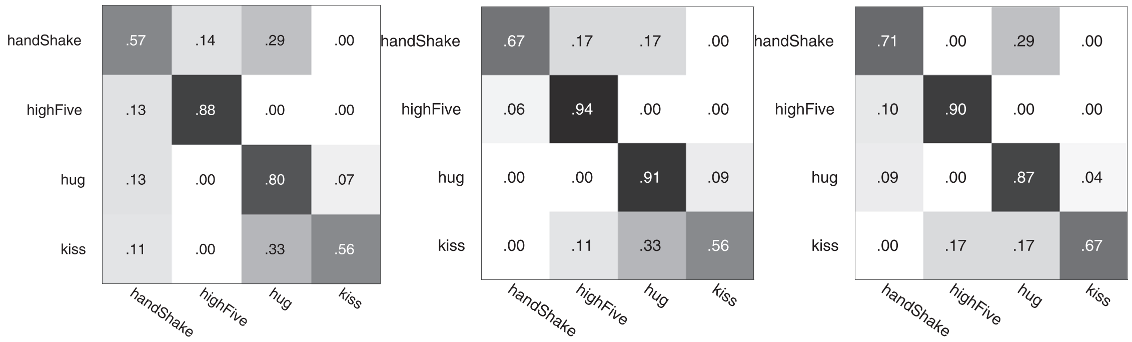
Method	mean AP
Our fusion	**70.7**
Our clustering	65.4
Fisher vector baseline	66.7
Wang et al. (2015a)	66.8
Jain et al. (2013)	62.5
Zhu et al. (2013)	61.4
Mathe and Sminchisescu (2012)	61.0
Jiang et al. (2012)	59.5
Gaidon et al. (2013)	54.0

Accuracy in a multi-label setting can be measured, as suggested in Madjarov et al. (2012), using per example accuracy. For an example x from a test dataset $\mathcal{D}$, we define $\mathcal{L}_x$ as the set of ground truth labels. For a classifier $h(\,\cdot\,)$, $h(x)$ is the set of predicted labels for example x. The accuracy, averaged on all samples is

$$\text{accuracy} = \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}}\frac{|h(x)\cap\mathcal{L}_x|}{|h(x)\cup\mathcal{L}_x|} \tag{21}$$

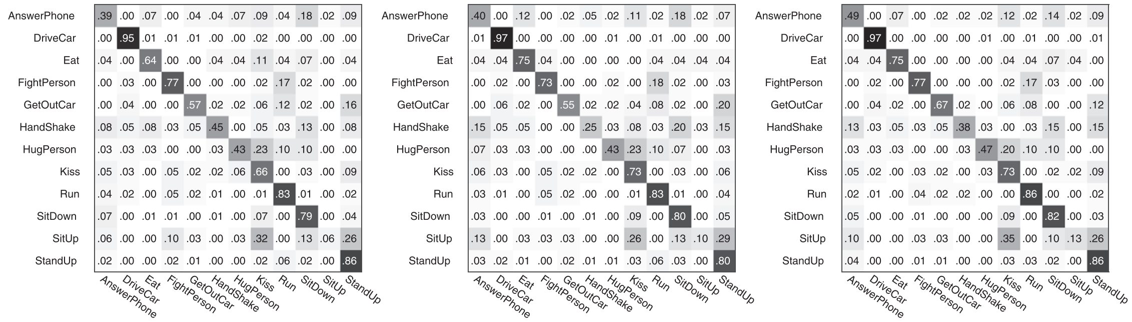
We obtain $h(x)$, after hard thresholding 1-vs-all classifier decisions. We report 61.3% for fusion, 57.7% for clustering and 60.0% for Fisher Vector baseline label-set accuracy. These results are consistent with mAP, which also takes into account multiple labels by evaluating ranking.

In Table 6 we report a comparison on the HighFive dataset; we report 77.6% mAP using our method with fusion. On this dataset the clustering approach performs already much better than our baseline, as also shown in Fig. 9. Fusion performs comparably to clustering improving only by 0.5%. HighFive is focused on human



(a) Baseline (b) Our Clustering (c) Our Fusion

Fig. 9. Confusion matrices for highfive. Fusion is able to improve recognition of "handShake" and "kiss", which are often misclassified in our baseline and in clustering approach. However, recognition of "highFive" and "hug" is slightly lower than in the clustering configuration. Mean per-class accuracy is respectively: 70.2%, 77.0% and 78.8%.



(a) Baseline (b) Our Clustering (c) Our Fusion

Fig. 10. Confusion matrices for Hollywood2, computed for single-label videos. Fusion improve most of the classes. Mean per-class accuracy is respectively: 61.6%, 61.2% and 65.7%.

Table 6
Comparison with the state of the art on the HighFive dataset. Results are reported as mean average precision over the 4 classes.

Method	mean AP
Our fusion	**77.6**
Our clustering	77.1
Fisher vector baseline	68.5
Wang et al. (2015a)	69.4
Karaman et al. (2014)	65.4
Ma et al. (2013)	53.3
Gaidon et al. (2013)	55.6
Laptev et al. (2008)	36.9
Patron-Perez et al. (2012)	42.4

interaction and, since it is collected from TV shows, clips usually depict actors in the foreground. Our clustering approach is able to match important patterns like hands shaking or faces approaching obtaining state-of-the art performance. Global representations are probably less effective since global information is not a strong cue for the actions. Moreover a global pooling approach is likely to gather some noise from actors not involved in the interaction to be recognized.

Note that the best performing, previously published method, on both datasets by Wang et al. (2015a), is using a person detector to improve the optical flow based frame stabilization while, to keep our method more generic, we do not rely on such technique in the extraction of dense trajectories. Also note that on both Hollywood2 and HighFive datasets our Fisher Vector implementation has a high performance. We believe that such figures are reached employing all of IDT based descriptors and adding the trajectory local coordinates to the descriptors before encoding.

6.3. Cluster saliency

To gain insight on the potential localization capability of our approach, we show a method for salient cluster mining. We would like to find which are the clusters that better help the classifier to discriminate actions. Given a video feature set $\boldsymbol{Z}$, the learned kernel SVM classifier for an action (defined by Eq. (13)), the weights α_k and training sample labels y_k we greedily search for the cluster $\boldsymbol{Z}_i$ that, if removed, causes the higher classification score drop on a correctly classified action:

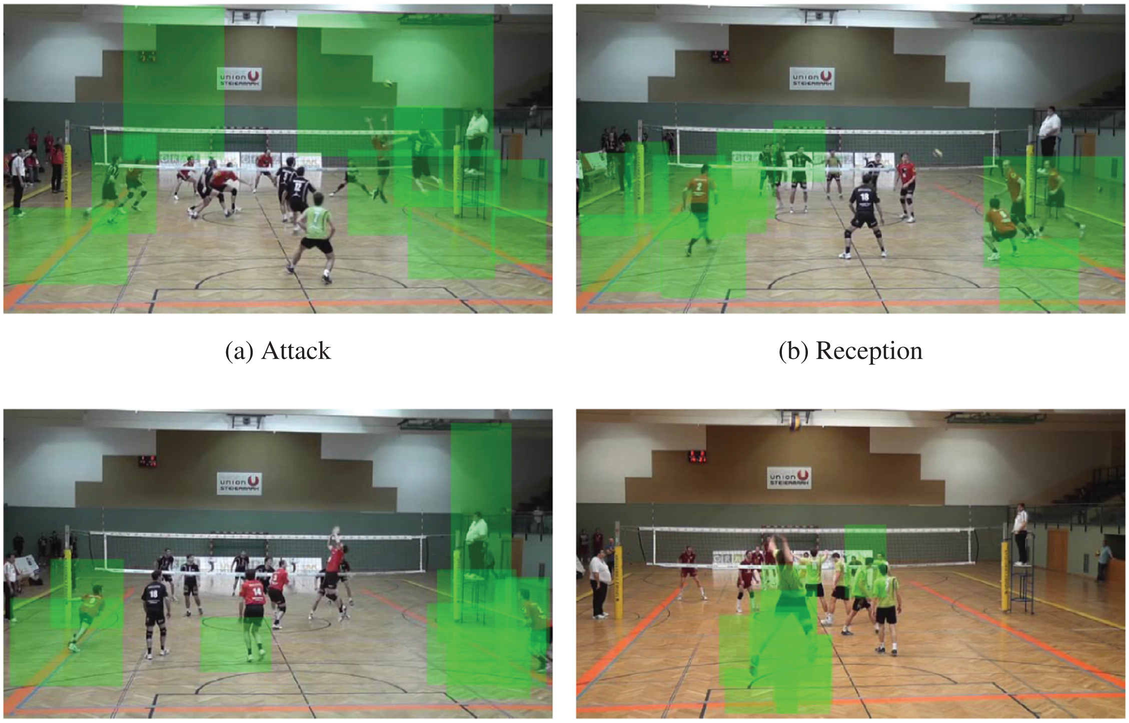
$$\boldsymbol{Z}_i = \arg\max_i \sum_k \alpha_k y_k \left[K_{\boldsymbol{X}^k}(\mathcal{P}(\boldsymbol{Z})) - K_{\boldsymbol{X}^k}(\mathcal{P}(\boldsymbol{Z}) \setminus Z_i) \right] \tag{22}$$

where $K_{\boldsymbol{X}}(\boldsymbol{Z}) = K(\boldsymbol{X}, \boldsymbol{Z})$.

Salient clusters are selected iteratively, removing them from the clip using Eq. (22) until the classification score does not decrease anymore.

We show some results for the Volleyball Activity dataset. We use volleyball as a test since volleyball plays are very well structured with precise running path and positions for players. For the sake of visualization, we represent each salient cluster as a bounding box, defined as the one containing all cluster features. In Fig. 11 we plot the accumulation of salient clusters bounding boxes, generating a heat map, which highlights the most relevant areas in the scene for the action.

It can be seen that for the "Service" and "Attack" action the serving and attacking players are respectively highlighted, while in the examples of "Reception" and "Setting" multiple players


(a) Attack

(b) Reception

(c) Setting

(d) Service

Fig. 11. Heatmaps of the most relevant clusters for collective actions “Attack”, “Reception”, “Service” and “Setting”. Heatmaps highlight motion of players that our classifier find most discriminative.

are highlighted. In the “Setting” action, both spikers, the middle-blocker and the opposite player run-up are localized.

This behaviour can be expected, as some actions need contextual information to be recognized. It may also happen that some actions are correctly classified by our clustering approach, but applying Eq. (22) results in highlighting as meaningful some parts of the scene which are not intuitively descriptive of the action. This happens, for example, in the “Setting” class, where most salient areas correspond with the players which are preparing to attack, and not with the player who is actually setting the ball.

Although this experiment only provides some anecdotal evidence it shows how the learned classifier interpret the spatio-temporal structure of the video generated by the trajectory groupings. Unfortunately, as explained before, this form of saliency not always accumulates on the exact region where the action is performed. This is easily explained by the fact that for the “Setting” action the contextual representation may be a stronger cue to recognize the activity.

Therefore if a precise action localization is sought we argue that a stronger supervisory signal must be provided. In the following Section, we show how the method discussed in Section 5, exploiting CNN and bounding box annotations allows precise localization of actions in space and time.

6.4. Action localization

In this section we evaluate the localization method presented in Section 5. We test our approach on UCF Sports which presents many challenges in the localization task. The actions present in UCF Sports clips have a high variation in body pose and motion, spanning to very static actions such as “Golfing” to highly dynamic and acrobatic sequences such as “Swing-Bench” which are extracted from pommel horse competitions.

We adhere to the same protocol of Tian et al. (2013), Lan et al. (2011) evaluating jointly classification and localization, using the Receiver Operating Characteristic (ROC) curve. Each frame is annotated with a single ground truth action bounding box. ROC curves are computed by plotting True Positive Rate (TPR) against the False Positive Rate (FPR), by varying some criterion. In our case we use the classifier output as criterion, meaning that positive samples should have higher scores than negative ones.

To compute the ROC curve we consider a video sample a True Positive only if it is correctly classified and the Intersection over Union of the predicted localizations with the ground truth, averaged on the whole clip, is above a given threshold σ.

As a synthetic measure of ROC evaluated systems, the Area Under Curve (AUC) is often used. AUC can also be interpreted as the probability of a random positive data sample to have a higher score than a negative one, therefore higher AUC means a better performing system.

We report AUC varying σ from 0.1 to 0.5 comparing the result to Tian et al. (2013), Lan et al. (2011). Note that, following the protocol proposed by Lan et al., ROC curves are computed limiting the False Positive Rate to 0.6, for this reason a perfect ROC will report an AUC of 0.6 instead of 1.0.

Compared with fully-supervised, tracking based methods (Gkioxari and Malik, 2015; Weinzaepfel et al., 2015) we perform worse. These methods are specifically developed to address localization and use tracking or a global tracklet optimization to obtain the final action spatio-temporal position.

Compared to other previously published results we obtain a much higher AUC for all thresholds except for 0.5. We have slightly lower performance with respect to Dynamic Poselets from Wang et al. (2014) for IoU=0.5, and we perform closer but better than Jain et al. (2014) for all thresholds.

In Fig. 12 we show some localizations obtained with our method. Our approach is able to deal with challenging lighting

Please cite this article as: F. Turchini et al., Understanding and localizing activities from correspondences of clustered trajectories, Computer Vision and Image Understanding (2016), http://dx.doi.org/10.1016/j.cviu.2016.11.007

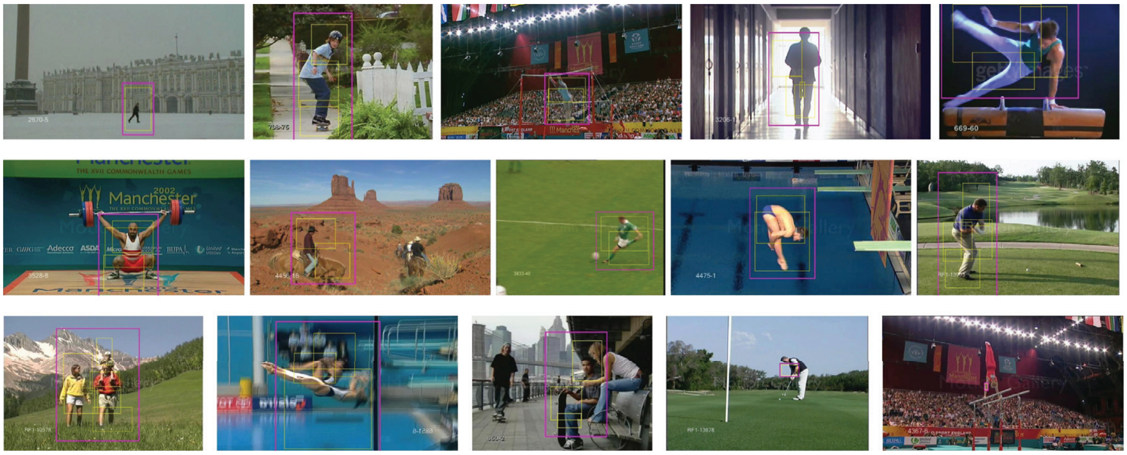


Fig. 12. Localization output of our algorithm. The yellow boxes are obtained from clusters and the magenta box is the final prediction. The first two rows report correct action localization, while the third row reports failure cases. Errors happen when multiple subjects performs the same action or in frames with little motion. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

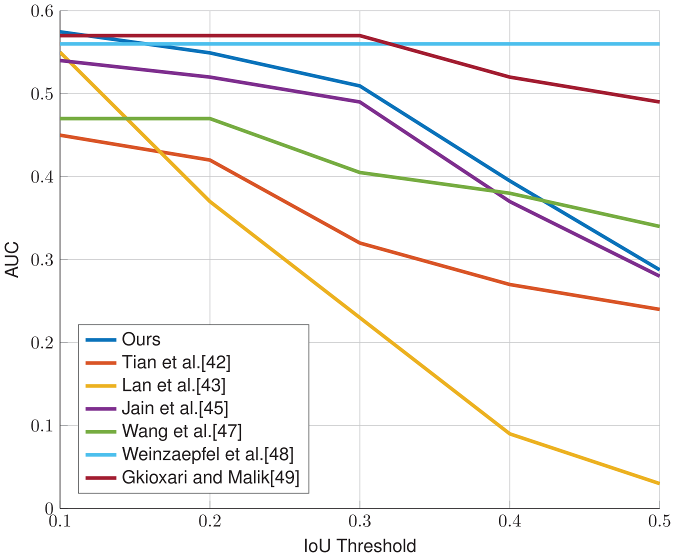


Fig. 13. Area under the ROC curve computed for different Intersection over Union (IoU) thresholds on UCF Sports dataset.

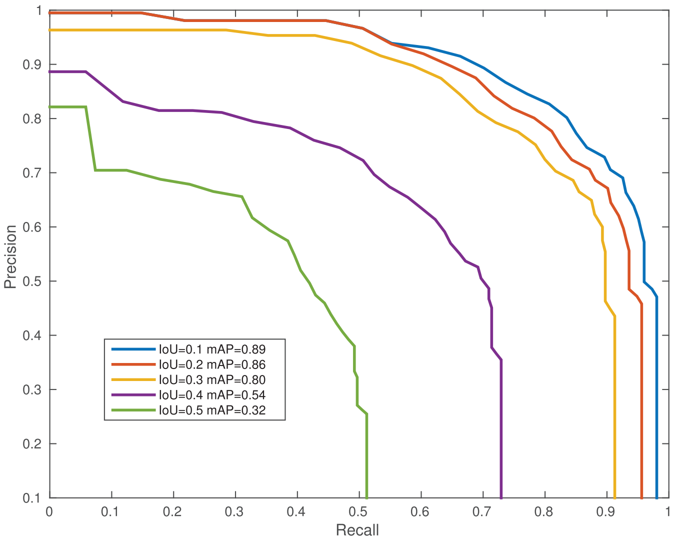


Fig. 14. Precision/Recall curve for action localization task on UCF sports, for different IoU thresholds. We report mean Aveage Precision for every IoU threshold in the legend.

conditions, high variation in scale and extreme human body poses. The third rows shows some localization errors. When people perform the same action our method may generate a single proposal when subjects are close, as in the first and second frame. In the third frame, a person sit handling a board is mistakenly predicted instead of the actual subject standing on the board in the background. Finally, for frames with little motion, the reduced amount of trajectories may not allow to generate correct action proposals, as in the last two examples.

The result reported in Fig. 13 agrees with the high performance reported on classification accuracy in Table 2. For lower, permissive, σ values, the classification quality prevails over the localization accuracy. Rising the IoU threshold allows to evaluate how precise a localization method is in predicting the location of the person performing the activity. We note that our method performs much better than Dynamic Poselets for all thresholds except for the stricter value of 0.5, for which we report a comparable, but lower, result. Consider that having an average IoU of 0.5 is a very strict setting, meaning that actions are correctly detected, according to PASCAL VOC2007 evaluation procedure for object detection, in every frame of the clip.

Note that both SDPM and Dynamic Poselets are specifically trained to perform action detection with a multi-part sliding window approach while our method is a simple byproduct of the video representation described in Section 3, scoring only a few proposal per frame with a simple linear classifier. Moreover, Dynamic Poselets relies on time consuming body joint annotations in every training frame while our method only requires persons bounding boxes to perform localization.

Finally it has to be noted that our classification accuracy is much better than the one reported in Tian et al. (2013),Lan et al. (2011), Jain et al. (2014),Weinzaepfel et al. (2015). We remark that our method addresses classification as the main task and localization is obtained thanks to the locality property of the unsupervised trajectory grouping.

In Fig. 14, we also report Precision/Recall curves for our approach for different IoU threshold values. As in the evaluation of AUC, we consider a video a True Positive, if correctly classified and the average IoU is above a threshold. Note that perfect recall may not be reached, in case the generated action proposals do not have a sufficient IoU with ground truth bounding boxes.

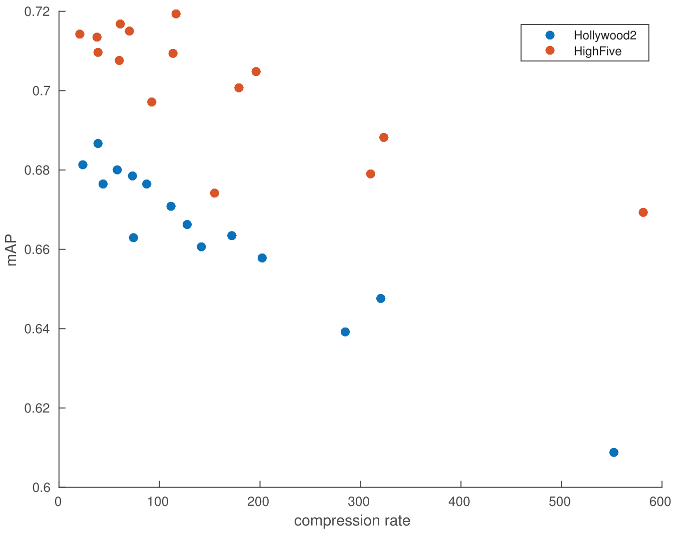


Fig. 15. Scatter plot of mAP and compression rate for several configurations on Hollywood2 and HighFive. In both datasets, mAP is inversely proportional to compression rate.

Table 7
Comparison of mean Average Precision for quantized and unquantized kernel on HighFive and Hollywood2 datasets using G=32, b=16. We report results for Our Clustering configuration and for Our Fusion.

Method	Dataset	Unquantized	Quantized
Our fusion	HighFive	76.1	75.9
Our clustering	HighFive	75.9	71.8
Our fusion	Hollywood2	66.7	65.1
Our clustering	Hollywood2	65.4	60.9

6.5. *Quantized kernel evaluation*

We evaluate the compression efficiency of our quantized representation by measuring the rate between the memory required to store and process the quantized version of clusters feature vectors and their unquantized counterpart. Product quantization is necessary when dealing with datasets with a large set of features like Hollywood2.

We first run a set of experiments to show how PQ affects performance in different settings on Hollywood2 and HighFive. We evaluate the mAP of our method using 10 clusters per video against the compression rate. It can be seen that mAP degrades for PQ configurations that attain higher compression rates. To give a better insight on the behaviour of quantized features, we report a scatter plot of mAP and compression rate for both datasets in Fig. 15. On both datasets compression rate is inversely proportional to mean Average Precision.

Finally in Table 7 we show how product quantization affect the mAP on HighFive and Hollywood2 datasets both for Clustering and Fusion approaches. We selected the parameters G and b that have a good trade-off between compression rate and mAP decrease.

Clustering only version of our method suffers more from the PQ process with respect to the Fisher Vector baseline. This can be explained by the fact that compressing many feature vectors will certainly accumulate more quantization errors. Using fusion we are able to compensate errors, attaining a minimum loss in mAP on both datasets.

Quantization of local features extracted with IDT, is not beneficial and leads to degraded performance. We ran an experiment to validate this assumption on HighFive using 256 Gaussians and 10 Clusters. We applied PQ to local features and kept the final cluster representations unquantized.

The best result that we obtained quantizing all features, yields 66.3 mAP while if we avoid quantizing the trajectory descriptors, we can obtain a 73.9 mAP. Compared to the completely unquantized result reported in Table 7, this is 2 mAP points below.

7. Conclusions

We have proposed a novel method for activity recognition based on local trajectory grouping and matching. Our approach allows to automatically understand what activities are performed in a video. Thanks to our cluster set kernel we can compute partial video correspondences effectively without exhaustively matching all local features.

The proposed method is extremely general and not tailored to specific sports or video shooting setting. Indeed our approach proves effective in recognizing activities both from Volleybal and Soccer broadcasts. Streams of these sports present very different viewpoints, player scales and cardinalities.

The good performance on this very diverse sports suggests that the proposed system may perform accurately also on generic activities. Therefore we ran several experiments on challenging generic actions datasets providing experimental evidence of the generality of the proposed approach, reporting state-of-the-art results. The experimental evaluation of our method showed that the performance is stable for parameters like the amount of Gaussians or the number of clusters per video.

The only drawback of our method is the need to compute, store and process multiple high-dimensional feature vectors per video. We deal with this issue formulating a quantized version of our kernel implementing a strong feature compression with little loss in recognition performance.

Finally given the local nature of our clusters we show how we can exploit clusters as natural action proposals to jointly recognize and localize activities in space and time. Adding frame-wise features improve the accuracy in classification and allows to train action localizers. Differently from exhaustive search action detectors we chain a localizer after the action recognition step. This allows us to evaluate few proposals per frame and still obtain good localization performance.

Acknowledgments

Lorenzo Seidenari and Francesco Turchini are partially supported by THE SOCIAL MUSEUM AND SMART TOURISM, MIUR project no. CTN010003423154 SMST.

References

Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C., 2014. Good practice in large-scale learning for image classification. IEEE Trans. Pattern Anal. Mach. Intell. 36 (3), 507–520.
Atmosukarto, I., Ghanem, B., Ahuja, S., Muthuswamy, K., Ahuja, N., 2013. Automatic recognition of offensive team formation in american football plays. In: Proceedings of Computer Vision in Sports, Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE.
Ballan, L., Bertini, M., Del Bimbo, A., Serra, G., 2010. Video event classification using string kernels. Multimed. Tools Appl. 48 (1), 69–87.
Bialkowski, A., Lucey, P., Carr, P., Denman, S., Matthews, I., Sridharan, S., 2013. Recognising team activities from noisy data. In: Proceedings of Computer Vision in Sports, Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE.
Bo, L., Sminchisescu, C., 2009. Efficient match kernel between sets of features for visual recognition. In: Advances in Neural Information Processing Systems, pp. 135–143.
Brun, L., Percannella, G., Saggese, A., Vento, M., 2016. Action recognition by using kernels on Aclets sequences. Comput. Vis. Image Underst. 144, 3–13.
Bruni, M., Uricchio, T., Seidenari, L., Del Bimbo, A., 2016. Do textual descriptions help action recognition? In: Proceedings of the 2016 ACM on Multimedia Conference. ACM, pp. 645–649.

Chatfield, K., Lempitsky, V.S., Vedaldi, A., Zisserman, A., 2011. The devil is in the details: an evaluation of recent feature encoding methods.. In: Proceedings of BMVC, 2, p. 8.

Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Return of the devil in the details: delving deep into convolutional nets. arXiv:1405.3531.

Chen, X., Cai, D., 2011. Large scale spectral clustering with landmark-based representation.. In: Proceedings of AAAI.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. 88 (2), 303–338.

Feragen, A., Kasenburg, N., Petersen, J., de Bruijne, M., Borgwardt, K., 2013. Scalable kernels for graphs with continuous attributes. Advances in Neural Information Processing Systems (NIPS).

Gade, R., Moeslund, T.B., 2013. Sports type classification using signature heatmaps. In: Proceedings of Computer Vision in Sports, Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE.

Gaidon, A., Harchaoui, Z., Schmid, C., 2013. Activity representation with motion hierarchies. Int. J. Comput. Vis. 1–20.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 580–587.

Gkioxari, G., Malik, J., 2015. Finding action tubes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 759–768.

Hsu, C.-C., Chen, H.-T., Chou, C.-L., Ho, C.-P., Lee, S.-Y., 2014. Trajectory based jump pattern recognition in broadcast volleyball videos. In: Proceedings of International Conference on Multimedia. ACM, New York, NY, USA, pp. 1117–1120. doi:10.1145/2647868.2654985.

Jain, M., Gemert, J., Jégou, H., Bouthemy, P., Snoek, C., 2014. Action localization with tubelets from motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 740–747.

Jain, M., Jégou, H., Bouthemy, P., 2013. Better exploiting motion for better action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2555–2562.

Jegou, H., Douze, M., Schmid, C., 2011. Product quantization for nearest neighbor search. IEEE Pattern Anal. Mach. Intell. 33 (1), 117–128.

Jégou, H., Douze, M., Schmid, C., Pérez, P., 2010. Aggregating local descriptors into a compact image representation. In: Proceedings of 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 3304–3311.

Jiang, Y.-G., Dai, Q., Xue, X., Liu, W., Ngo, C.-W., 2012. Trajectory-based modeling of human actions with motion reference points. In: Proceedings of Computer Vision–ECCV 2012. Springer, pp. 425–438.

Karaman, S., Seidenari, L., Ma, S., Del Bimbo, A., Sclaroff, S., 2014. Adaptive structured pooling for action recognition. In: Proceedings of the British Machine Vision Conference (BMVC). BMVA Press.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Kläser, A., 2010. Learning Human Actions in Video. Université de Grenoble Ph.D. thesis. URL: http://lear.inrialpes.fr/pubs/2010/Kla10

Kosmopoulos, D.I., Doulamis, N.D., Voulodimos, A.S., 2012. Bayesian filter based behavior recognition in workflows allowing for user feedback. Comput. Vis. Image Underst. 116 (3), 422–434.

Kovashka, A., Grauman, K., 2010. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: Proceedings of Computer Vision Pattern Recognition (CVPR). IEEE.

Kviatkovsky, I., Rivlin, E., Shimshoni, I., 2014. Online action recognition using covariance of shape and motion. Comput. Vis. Image Underst. 129, 15–26.

Lan, T., Wang, Y., Mori, G., 2011. Discriminative figure-centric models for joint action localization and recognition. In: Proceedings of International Conference on Computer Vision (ICCV). IEEE.

Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B., 2008. Learning realistic human actions from movies. In: Proceedings of Computer Vision and Pattern Recognition (CVPR). IEEE.

Liu, J., Carr, P., Collins, R.T., Liu, Y., 2013. Tracking sports players with context-conditioned motion models (oral). In: Proceedings of Computer Vision and Pattern Recognition (CVPR).

Ma, S., Zhang, J., Ikizler-Cinbis, N., Sclaroff, S., 2013. Action recognition and localization by hierarchical space-time segments. In: Proceedings of International Conference on Computer Vision (ICCV). IEEE.

Madjarov, G., Kocev, D., Gjorgjevikj, D., Dzeroski, S., 2012. An extensive experimental comparison of methods for multi-label learning. Pattern Recognit 45 (9), 3084–3104.

Mathe, S., Sminchisescu, C., 2012. Dynamic eye movement datasets and learnt saliency models for visual action recognition. In: Proceedings of the Computer Vision–ECCV 2012. Springer, pp. 842–856.

Niebles, J.C., Chen, C.-W., Fei-Fei, L., 2010. Modeling temporal structure of decomposable motion segments for activity classification. In: Proceedings of European Conference on Computer Vision (ECCV). Springer.

Oneata, D., Verbeek, J., Schmid, C., 2013. Action and Event Recognition with Fisher Vectors on a Compact Feature Set. In: Proceedings of International Conference on Computer Vision (ICCV). IEEE.

Patron-Perez, A., Marszalek, M., Reid, I., Zisserman, A., 2012. Structured learning of human interactions in tv shows. IEEE Trans. Pattern Anal. Mach. Intell. 34 (12), 2441–2453.

Perronnin, F., Sanchez, J., Mensink, T., 2010. Improving the fisher kernel for large-scale image classification. In: Proceedings of European Conference on Computer Vision (ECCV).

Raptis, M., Kokkinos, I., Soatto, S., 2012. Discovering discriminative action parts from mid-level video representations. In: Proceedings of Computer Vision and Pattern Recognition (CVPR). IEEE.

Ravanbakhsh, M., Mousavi, H., Rastegari, M., Murino, V., Davis, L.S., 2015. Action recognition with image based cnn features. arXiv preprint arXiv:1512.03980.

Revaud, J., Douze, M., Schmid, C., Jégou, H., 2013. Event retrieval in large video collections with circulant temporal encoding. In: Proceedings of Computer Vision and Pattern Recognition (CVPR). IEEE.

Rodriguez, M.D., Ahmed, J., Shah, M., 2008. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In: Proceedings of Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 1–8.

Roshtkhari, M.J., Levine, M.D., 2013. Online dominant and anomalous behavior detection in videos. In: Proceedings of Computer Vision and Pattern Recognition (CVPR). IEEE.

Ryoo, M., 2011. Human activity prediction: Early recognition of ongoing activities from streaming videos. In: Proceedings of International Conference on Computer Vision (ICCV). IEEE.

Sánchez, J., Perronnin, F., 2011. High-dimensional signature compression for large-scale image classification. In: Proceedings of 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 1665–1672.

Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J., 2013. Image classification with the fisher vector: theory and practice. Int. J. Comput. Vis. 105 (3), 222–245.

Seidenari, L., Serra, G., Bagdanov, A.D., Del Bimbo, A., 2014. Local pyramidal descriptors for image recognition. IEEE Trans. Pattern Anal. Mach. Intell. 36 (5), 1033–1040.

Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S., 2014. CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813.

Soomro, K., Zamir, A.R., 2014. Action recognition in realistic sports videos. In: Computer Vision in Sports. Springer, pp. 181–208.

Tian, Y., Sukthankar, R., Shah, M., 2013. Spatiotemporal deformable part models for action detection. In: Proceedings of Computer Vision and Pattern Recognition (CVPR). IEEE.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M., 2015. Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, pp. 4489–4497.

Turchini, F., Seidenari, L., Del Bimbo, A., 2015. Understanding sport activities from correspondences of clustered trajectories. In: Proceedings of ICCV International Workshop on Computer Vision in Sports (CVSports). Santiago, Chile.

Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W., 2013. Selective search for object recognition. Int. J. Comput. Vis. 104 (2), 154–171.

Vrigkas, M., Karavasilis, V., Nikou, C., Kakadiaris, I.A., 2014. Matching mixtures of curves for human action recognition. Comput. Vis. Image Underst. 119, 27–40.

Wallraven, C., Caputo, B., Graf, A., 2003. Recognition with local features: the kernel recipe. In: Proceedings of International Conference on Computer Vision (ICCV). IEEE.

Waltner, G., Mauthner, T., Bischof, H., 2014. Indoor activity detection and recognition for automated sport games analysis. In: Proceedings of the Workshop of the Austrian Association for Pattern Recognition (AAPR/OAGM).

Wang, H., Kläser, A., Schmid, C., Liu, C.-L., 2013. Dense trajectories and motion boundary descriptors for action recognition. Int. J. Comput. Vis. 60–79.

Wang, H., Oneata, D., Verbeek, J., Schmid, C., 2015a. A robust and efficient video representation for action recognition. Int. J. Comput. Vis. 1–20.

Wang, H., Schmid, C., 2013. Action recognition with improved trajectories. In: Proceedings of International Conference on Computer Vision (ICCV). IEEE, pp. 3551–3558.

Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C., et al., 2009. Evaluation of local spatio-temporal features for action recognition. In: Proceedings of British Machine Vision Conference (BMVC).

Wang, J., Liu, Z., Wu, Y., Yuan, J., 2012. Mining actionlet ensemble for action recognition with depth cameras. In: Proceedings of Computer Vision and Pattern Recognition (CVPR). IEEE.

Wang, L., Qiao, Y., Tang, X., 2014. Video action detection with relational dynamic–poselets. In: Proceedings ofComputer Vision–ECCV 2014. Springer, pp. 565–580.

Wang, L., Qiao, Y., Tang, X., 2015b. Action recognition with trajectory-pooled deep–convolutional descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4305–4314.

Weinzaepfel, P., Harchaoui, Z., Schmid, C., 2015. Learning to track for spatio-temporal action localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3164–3172.

Yeffet, L., Wolf, L., 2009. Local trinary patterns for human action recognition. In: Proc. of International Conference on Computer Vision (ICCV). IEEE.

Yu, G., Yuan, J., 2015. Fast action proposals for human action detection and search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1302–1311.

Zhu, J., Wang, B., Yang, X., Zhang, W., Tu, Z., 2013. Action recognition with actons. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3559–3566.