Chapter 9
Action Recognition in Realistic Sports Videos

Khurram Soomro and Amir R. Zamir

Abstract The ability to analyze the actions which occur in a video is essential for automatic understanding of sports. Action *localization* and *recognition* in videos are two main research topics in this context. In this chapter, we provide a detailed study of the prominent methods devised for these two tasks which yield superior results for sports videos. We adopt UCF Sports, which is a dataset of realistic sports videos collected from broadcast television channels, as our evaluation benchmark. First, we present an overview of UCF Sports along with comprehensive statistics of the techniques tested on this dataset as well as the evolution of their performance over time. To provide further details about the existing action *recognition* methods in this area, we decompose the action recognition framework into three main steps of feature extraction, dictionary learning to represent a video, and classification; we overview several successful techniques for each of these steps. We also overview the problem of *spatio-temporal localization* of actions and argue that, in general, it manifests a more challenging problem compared to action recognition. We study several recent methods for action localization which have shown promising results on sports videos. Finally, we discuss a number of forward-thinking insights drawn from overviewing the action recognition and localization methods. In particular, we argue that performing the recognition on *temporally untrimmed* videos and attempting to describe an action, instead of conducting a forced-choice classification, are essential for analyzing the human actions in a realistic environment.

K. Soomro (✉)
Center for Research in Computer Vision, University of Central Florida,
Orlando, FL 32826, USA
e-mail: ksoomro@cs.ucf.edu

A.R. Zamir
Gates Computer Science, #130, Stanford University, 353 Serra Mall, Stanford, CA 94305, USA
e-mail: zamir@cs.stanford.edu

T.B. Moeslund et al. (eds.), *Computer Vision in Sports*,
Advances in Computer Vision and Pattern Recognition,
DOI 10.1007/978-3-319-09396-3_9

9.1 Introduction

Developing automatic methods for analyzing actions in videos are of particular importance for machine understanding of sports. Action recognition, which is the problem of assigning a video to a set of predefined action classes, and action localization, defined as identification of the spatio-temporal location where an action takes place, are two of the fundamental and heavily studied topics in this context.

The majority of existing frameworks for action recognition consist of three main steps: feature extraction, dictionary learning to form a representation for a video based on the extracted features, and finally classification of the video using the representation. In the first step, a set of features, such as STIP [32, 33] or dense trajectories [77, 78], are extracted from a given video. These features are supposed to encode the information which is useful for recognition of the action in a numerical form such as a vector. Then, the extracted features are used for forming a representation of a video, which captures the actions that occur therein. Such representation may be as simple as a histogram of most frequent motions [32, 33, 75, 77, 78] or a semantically meaningful model such as action poses [76]. In the last step, a general model for each action of interest is learnt using the computed representation of a set of labeled training videos. Given the learnt models, a query video is then assigned to the most similar action class by the classifier.

Action localization, unlike action recognition, deals with the problem of identifying the exact location in space-time where an action occurs. It manifests a wider range of challenges, e.g., dealing with background clutter or the spatial complexity of the scene, and has been the topic of fewer research papers as compared to action recognition. The recent successful action localization techniques, which will be discussed in Sect. 9.4, typically attempt to segment the action utilizing cues based on the appearance, motion, or a combination of both [40, 69].

UCF Sports [57] is one of the earliest action recognition datasets that contains realistic actions in unconstrained environment. In this chapter, we provide a detailed study of the UCF Sports dataset along with comprehensive statistics of the methods evaluated on it. We also discuss the technical details of the prominent approaches for action localization and recognition which yield superior results for sports videos.

In Sect. 9.5, we discuss the insights acquired from summarizing the existing action recognition and localization methods, especially the ones evaluated on UCF Sports. In particular, we will argue that many of the existing action localization and recognition systems are devised for *temporally trimmed* videos. This is a significantly unrealistic assumption, and it is essential to develop techniques which are capable of performing the recognition or localization on temporally untrimmed videos. In addition, we discuss that describing an action using a universal lexicon of lower level actions, also known as action attributes [17, 37], is a worthwhile alternative to increasing the number of classes in action datasets and performing a forced-choice classification task. Lastly, we will discuss that the majority of existing action localization algorithms mostly perform an exhaustive search in, spatial, temporal, or spatio-temporal domain, in order to find a match for their action representation. This approach is

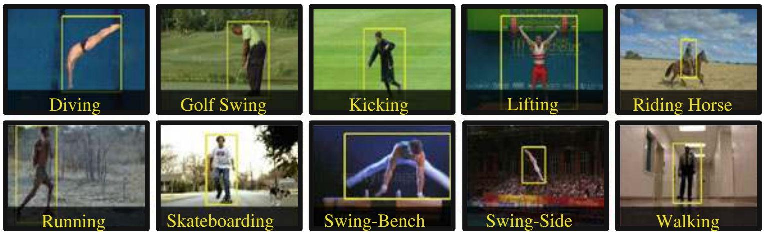

Fig. 9.1 UCF Sports Dataset: sample frames of 10 action classes along with their bounding box annotations of the humans shown in *yellow*

inefficient, as also observed in several recent object detection methods [9, 15, 72], and can be improved by employing a more efficient search strategy similar to selective search [25, 72] or object proposals [15].

9.2 UCF Sports Dataset

UCF Sports consists of various sports actions collected from broadcast television channels including ESPN and BBC. The dataset includes 10 actions: diving, golf swing, kicking, lifting, riding horse, running, skateboarding, swinging-bench, swinging-side, and walking. Figure 9.1 shows a sample frame of all ten actions. The dataset along with human bounding box and human gaze annotations is available to the public.[1]

The dataset includes a total of 150 sequences with the resolution of 720 × 480. Table 9.1 summarizes the characteristics of the dataset. Figure 9.2 shows the distribution of the number of clips per action as the number of clips in each class is not the same. Figure 9.3 illustrates the total duration of clips (blue) and the average clip length (green) for every action class. It is evident that certain actions are short in nature, such as kicking, as compared to walking or running, which are relatively longer and have more periodicity. However, it is apparent from the chart that the average duration of action clips shows great similarities across different classes. Therefore, merely considering the duration of one clip would not be enough for identifying the action.

Table 9.1 Summary of the characteristics of UCF Sports

Actions	10	Total duration	958 s
Clips	150	Frame rate	10 fps
Mean clip length	6.39 s	Resolution	720 × 480
Min clip length	2.20 s	Max num. of clips per class	22
Max clip length	14.40 s	Min num. of clips per class	6

[1] Download UCF Sports dataset: http://crcv.ucf.edu/data/UCF_Sports_Action.php.

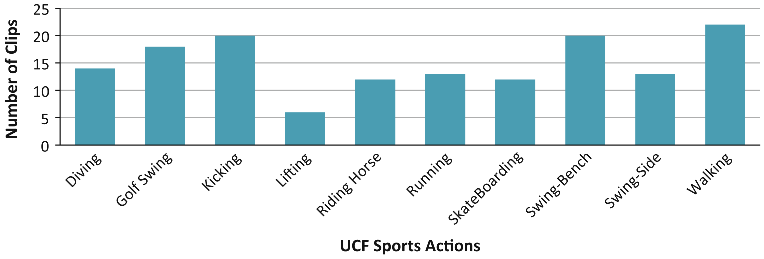

Fig. 9.2 Number of clips per action class

UCF Sports has been cited by more than 400 times since its introduction in 2008 and has been adopted by many state-of-the-art algorithms as an evaluation benchmark. Figure 9.4 shows number of citations per year and the cumulative count showing its rise every year.

Since its introduction, the dataset has been used for numerous applications such as: action recognition, action localization, and saliency detection. Saliency detection methods specify the region in the videos which attracts the human attention the most. The following sections explain how the methodologies applied on UCF Sports have evolved over time and describe the standard experimental setups. In the rest of this chapter, we focus on action recognition and localization as the two main tasks for analyzing the actions in videos.

9.2.1 Action Recognition

Action recognition is one of the heavily studied topics in computer vision. More than 80 % of the research publications which have utilized UCF Sports reported action recognition results on this dataset. Figure 9.5 shows how the accuracy of action recognition on UCF Sports has evolved every year since 2008. This figure reports yearly mean accuracy for two experimental setups: (1) Leave-One-Out and (2) Five-

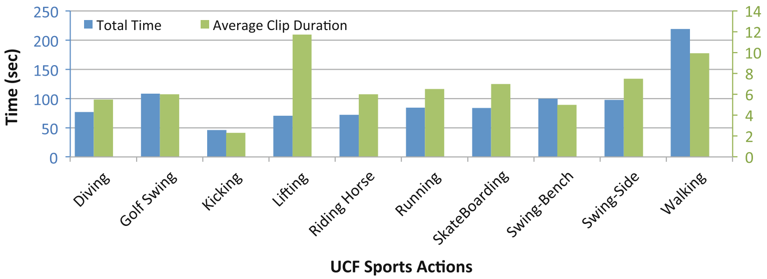

Fig. 9.3 The total time of video clips for each action class is shown in *blue*. Average length of clips for each action is shown in *green*

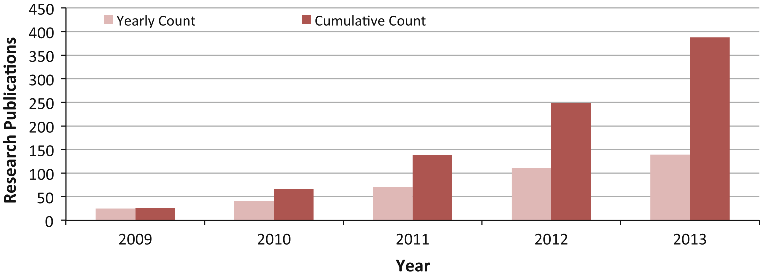

Fig. 9.4 Research publications in which UCF Sports was utilized. The chart shows yearly and cumulative counts until 2013

Fold cross-validation. The mean accuracy has gradually improved every year and the state-of-the-art method (96.6 % accuracy [23]) is able to recognize the minority of the actions correctly.

The early successful methods on UCF Sports were mostly based on sparse space-time interest points [32, 33] and cuboid descriptors. However, the more recent dense sampling [18, 50] and trajectory-based [77, 78] techniques have been able to outperform them and significantly improve the overall accuracy. Several of such methods for feature extraction [16, 32, 33], action representation [26, 76], dictionary learning [53], and classification [51, 57] will be discussed in more detail in Sect. 9.3.

9.2.1.1 Experimental Setup

The original way [57] to test on UCF Sports was to use a Leave-One-Out (LOO) cross-validation scheme. This scenario takes out one sample video for testing and trains using all of the remaining videos of an action class. This is performed for every sample video in a cyclic manner, and the overall accuracy is obtained by averaging the accuracy of all iterations.

An alternative experimental setup was proposed in [86] that uses a five-fold cross-validation scheme. Each fold consisted of one-fifth videos for testing and the remaining for training. For each fold the recognition accuracy is calculated using

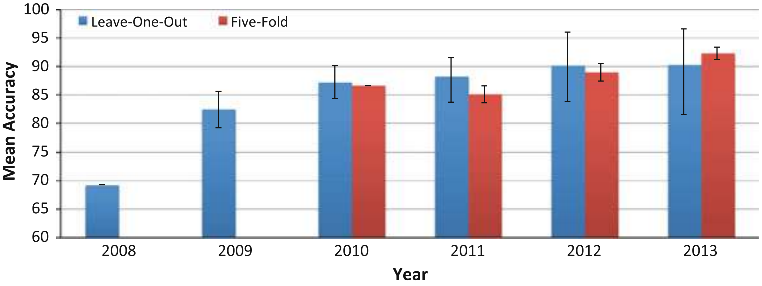

Fig. 9.5 Yearly mean accuracy of action recognition for two experimental setups: (1) Leave-One-Out (2) Five-Fold cross-validation. The brackets show the maximum and minimum accuracy for every year

average class accuracy and the results are averaged over all folds to get the final accuracy.

For action localization methods that perform action recognition as well, [31] proposed a different testing scheme that uses one-third videos for each action class as testing and the remaining two-thirds for training. Further details can be seen in Sect. 9.2.2.1.

All of the three aforementioned experimental setups of LOO [30, 35, 57, 58, 81, 87], fivefold cross-validation [53, 86], and predefined splits [31, 69] have been used for performing the evaluations on UCF Sports, while LOO and fivefold are more common. However, fivefold cross-validation and using splits are computationally less expensive as they require less training-testing iterations.

9.2.2 Action Localization

Action localization is an important problem with a wide range of applications such as human activity analysis, behavior recognition, video retrieval, and many others. Action localization is generally a more difficult task compared to action recognition. That is because a successful action localization requires the action class to be correctly recognized, and also its spatio-temporal location to be identified. Therefore, action recognition is a subproblem solved within action localization. In action recognition, at times it is feasible to recognize an action using only the background features, and without utilizing the temporal or motion information. Therefore, even though one could recognize such actions, detecting the exact spatio-temporal location of the action would require additional steps such as modeling the temporal movement of body parts. This becomes even more complex when practical challenges, e.g., occlusion or background clutter, are considered as well.

Action localization has been the subject of fewer research efforts compared to action recognition. This is observable in the statistics as less than 15 % of the research papers evaluated on UCF Sports have discussed action localization results. UCF Sports was one of the first datasets for which bounding box annotations of the actions were made available, and therefore it is an excellent dataset for studying the advances on this topic. Table 9.2 shows some of the recent action localization approaches and their results on this dataset.

Action localization approaches have mostly focused on developing action representation models which suit the problem of spatio-temporal localization. The figure-centric model [31] using the human location as a latent variable, Spatio-temporal Deformable Part Model (SDPM) [69], or hierarchical space-time segments [40] are some of the notable methods in this context. The key idea behind the majority of these methods is to capture the human structure and its deformity in a spatio-temporal framework. Further details of several localization methods tested on sports videos are provided in Sect. 9.4.

Table 9.2 Action localization results on UCF sports

Method	Year	Experimental setup	Evaluation metric	Accuracy (%)
Shapovalova et al. [61]	2013	Splits	ROC & AUC	32.3
Lan et al. [31]	2011	Splits	ROC & AUC	38
Tian et al. [69]	2013	Splits	ROC & AUC	42
Tran and Yuan [71]	2012	Splits	Average precision	55.34[a]
Ma et al. [40]	2013	Splits	Intersection-over-union	42.1
Gall et al. [21]	2011	Five-fold cross-validation	Average precision	54
Yao et al. [86]	2010	Five-fold cross-validation	Average precision	54
Cheng et al. [10]	2013	Five-fold cross-validation	Average precision	61.6
Thi et al. [68]	2012	Five-fold cross-validation	Binarized overlap	89
Raptis et al. [56]	2012	Leave-one-out cross-validation	Localization score	62.6

[a] Only three actions were used for localization: Horse Riding, Running and Diving

9.2.2.1 Experimental Setup

Similar to Sect. 9.2.1, the original experimental setup for action localization is to use Leave-One-Out (LOO) scheme. However, this setup has been criticized by Lan et al. [31] for two main reasons. The first reason is that no parameters (e.g., the SVM regularizer weightings, C) have been given and experiments show that the accuracy can change drastically by varying parameters. The second reason, which is more critical, is that many videos have similar backgrounds, and therefore a strong scene correlation exists between videos. Thus, while testing under LOO setting, the classifier may use the background features to classify the video which results in an artificially high accuracy. An empirical evidence for this issue has been provided in [31]. To alleviate this problem, an alternate experimental setup [31] is proposed. The new setup[2] splits the dataset into two uneven parts: two-third of videos for training and one-third for testing. To calculate the accuracy, an *intersection-over-union* criterion is used to plot ROC curves with a certain overlap threshold. The *intersection-over-union* computes the overlap between the predicted bounding box and the ground truth, and divides it by the union of both the bounding boxes, for every frame. This value is then averaged over all frames in a video. A 20 % overlap threshold is used for this experiment. Area Under Curve (AUC) against the overlap threshold, which shows how the performance varies if the threshold is changed, is used to compute the final performance. To calculate the overlap, the ground truth bounding box per frame is provided for the dataset. Figure 9.1 shows sample frames from UCF Sports dataset for each action class along with their annotated bounding boxes of humans.

The reported results in Table 9.2 show a variety of experimental setups and evaluation metrics. Due to the aforementioned issues, adopting the predefined splits of

[2] UCF Sports experimental setup for Action Localization: http://www.sfu.ca/~tla58/other/train_test_split.

[31] as the setup is recommended. As the evaluation metric, both Precision/Recall and ROC curves are appropriate. However, Precision/Recall has been a more popular choice for the other datasets, and therefore it is recommended for UCF Sports as well for the sake of consistency.

9.3 Action Recognition in Realistic Sports Videos

In this section, we provide an overview of some of the action recognition techniques which have shown superior results for sports videos. The overall recognition framework is broken down into three major steps of feature extraction, dictionary learning (for forming the representation of videos), and finally classification. The prominent methods for each step are elaborated in the rest of this section.

9.3.1 Feature Extraction

Classifying actions from videos requires extracting a set of data points, commonly termed features, that are expected to carry the information which is useful for distinguishing between various actions. Existing features can be classified into two main categories: (1) Low-level (Local) and (2) High-level (Holistic) features. Low-level features are the most commonly used features and are extracted by either detecting interest points or densely sampling them in a video. High-level features capture further structured information related to the action being performed. This high-level structure aims at gathering features such as shape [26], pose [76] or contextual information [81]. The general goal of feature extraction is to gather features that are generic enough and robust to backround variation. These features should be invariant to changes in scale, rotation, affine transformations, illumination, and viewpoint. However, capturing the required information while preserving the robustness to the aforementioned issues is a challenging problem. In the following sections, we will explore several low-level and high-level features that have performed well on UCF Sports.

9.3.1.1 Low-Level Features

Generally, low-level feature extraction is done by detecting sparse keypoints such as corners [24, 62], edges [8, 48], contours [49, 52], or blobs [42]. The corner detectors usually work by finding the locations that have large gradients in all directions; to find edges, an intensity derivative is applied in a specific direction. Contours find local silhouettes, and blobs aim at detecting regions within an image that have distinctive color properties compared to the neighboring areas. Once the keypoints are detected, a descriptor is formed around the point, which captures the local information. This descriptor can be scale-invariant such as Scale-Invariant Feature Transform

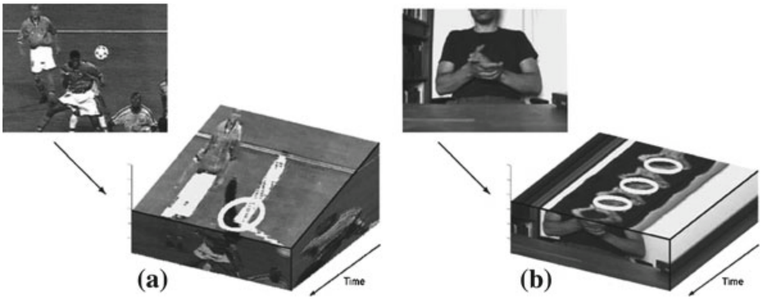

Fig. 9.6 Space-Time Interest Points (STIP) [32]: Features detected in **a** Football sequence when player is heading a ball. **b** Hand clapping sequence. The temporal slice of space-time volume shows that the detected events correspond to neighborhoods with high spatio-temporal variation

(SIFT) [38], texture-based [54], shape-context [5], gradient location and orientation histogram (GLOH) [46], steerable filters using Gaussian derivatives [20], moment invariants [73], and many more. Other useful information that local features can extract could be color, motion, trajectories, shape descriptors, and depth cues.

The aforementioned features were initially devised for images and capture the static information from the image. To extend this to videos and make use of temporal information, Space-Time Interest Points (STIP) [32, 33] were developed which are based on an extension of Harris corner detector. STIP can be described as the extention of spatial interest points into the spatio-temporal domain, where local structures which have significant local variations of image intensity values in space and nonconstant motion in time are detected. These local spatio-temporal features often correspond to interesting events in video data (see Fig. 9.6). Inspired by STIP, many variants have been proposed over time such as 3D-SIFT [60], HOG3D [29], extended SURF [80], and Local Trinary Patterns [87].

Although sparse keypoints have shown good classification performance, dense sampling has given further improvements in the image classification task [18, 50]. Dense sampling is different from sparse interest points in the sense that points are uniformly selected from the image, instead of using a criteria for selecting keypoints. That way, more information is gathered which can be learned by a classifier to give better performances in action recognition.

A more intuitive way of finding spatio-temporal characteristics is to track interest points throughout the video sequence. Some recent methods [43, 45, 65, 66] have shown improvement in action recognition performance using motion information of trajectories. These methods obtain feature trajectories by either using KLT tracker [39] in their approach [43, 45] or matching SIFT descriptors [38] between consecutive frames [66].

In the following sections, we describe two main low-level features that have given superior action recognition results on the UCF Sports dataset.

Color Space-Time Interest Points

An extension of STIP [32] to Color-STIP [16] has been proposed recently which gives the best performance in action recognition, compared to other sparse space-time interest point features. This approach uses a multichannel reformulation of existing STIP detectors and descriptors by considering different chromatic representations derived from opponent color space. Chromaticity specifies the quality of a color and consists of two parameters: hue and saturation. Adding Color to the temporal domain allows for better motion estimation and temporal variation.

The approach transforms *RGB* space to *Opponent* color space and comes up with a new photometric representation: I(ntensity), C(hromatic), N(ormalized chromatic) and H(ue). A combination of intensity and chromatic channels is used to give a three-channel representation: *IC*, *IN*, and *IH*. Multi-Channel Harris-STIP and Gabor-STIP detectors are formulated for each photometric channel and are represented using a Multi-Channel STIP descriptor. The Multi-Channel STIP descriptor is calculated by incorporating chromaticity in the HOG3D [29] descriptor. The final descriptor is a combination of two variants: (1) Channel Integration and (2) Channel Concatenation.

Many existing STIP-based approaches [32, 33] operate on image intensity, making them sensitive to highlights and shadows. They ignore the discriminative information by discarding chromaticity from the representation. By utilizing chromaticity in their enhanced appearance model, Color-STIP has shown to outperform other STIP-based methods.

Trajectories

Dense sampling of feature points and feature trajectories [39] have shown a notable improvement in image classification [18, 50] and activity recognition [45]. Inspired by these approaches, a recent method [77] computes dense trajectories by sampling dense points from each frame and tracking them based on displacement information from a dense optical flow field. By employing global smoothness constraints, dense trajectories are made to be more robust to irregular abrupt motion as well as shot boundaries.

Feature points are densely sampled on a grid, with uniform spacing, and tracked at multiple spatial scales to obtain dense trajectories (see Fig. 9.7). Each point is tracked between consecutive frames using median filtering in a dense optical flow field. These points from subsequent frames are then linked together to form a trajectory. To avoid the problem of drifting in trajectories, the length is limited to a fixed number of frames. Drifting usually occurs when a trajectory moves away from the initial position during the tracking process.

In order to encode various trajectory information, a novel descriptor is proposed which combines trajectory shape, appearance, and motion information. The shape of trajectory embeds local motion patterns and is described by a sequence of normalized displacement vectors. The motion and appearance information is captured by computing the descriptor within a space-time volume aligned with the trajectory.

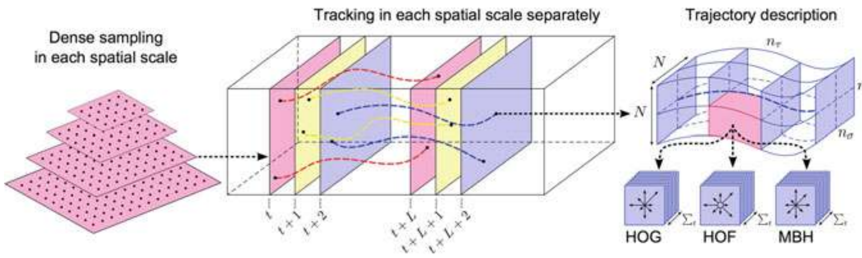

Fig. 9.7 Extracting Dense Trajectories [78]. *Left* Densely sampled feature points on each spatial scale. *Middle* Tracking feature points separately on each spatial scale using median filtering for L frames in a dense optical flow field. *Right* A $N \times N$ pixels neighborhood divided into $n_\sigma \times n_\sigma \times n_\tau$ grid is used to compute (*HOG*, *HOF*, *MBH*) descriptors along the trajectory

The volume is subdivided into a spatio-temporal grid to obtain local information. Histogram of Oriented Gradients (HOG) [11] encoding static appearance information, Histogram of Oriented Flow (HOF) [34] getting local motion information and Motion Boundary Histogram (MBH) [12] capturing relative motion information, are used as various descriptors for the trajectory volume.

Dense trajectories have shown to give better performance than Space-Time Interest Points (STIP) [33], as they capture appearance and dynamic motion information along the trajectory as compared to STIP, which uses cuboids instead. Experiments have shown dense trajectories to be further robust to camera motion and more informative for action classification.

9.3.1.2 High-Level Features

High-level features in action recognition represent an action by detecting high-level concepts [58] and often build upon local features [67]. The main idea is to preserve structural information of actions. These high-level features can be a spatio-temporal volume (STV) generated by 2D contours [88], 3D shapes induced by silhouettes [22], motion descriptor based on smoothed and aggregated optical flow [14], kinematic features [4] and so on.

The following sections introduce two of the recent high-level features that model the action remarkably well under varying background conditions and yield superior results on UCF Sports.

Spatio-Temporal Structures of Human Poses

An action can be considered as a mere articulation of parts. Hence, representing an action as poses is intuitively meaningful and has the capability of incorporating the variety of nonrigidness that a human body posesses when performing an action. In this context, [76] presents a pose-based action representation which models the spatio-temporal structures of human poses [85].

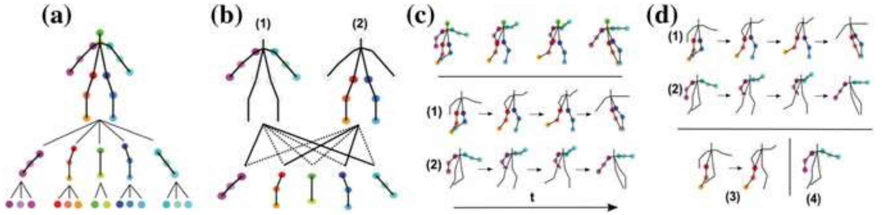

Fig. 9.8 Representing actions with poses [76]. **a** A pose consists of 14 joints combined together to form five body parts; **b** two spatial-part-sets showing co-occurring combinations of body parts for a particular action; **c** temporal-part-sets showing co-occurring sequences of body parts; **d** action representation consisting of spatial-part-sets (*4*) and temporal-part-sets (*1–3*)

Given a video, K-best pose estimations are obtained for each frame, and the best pose is inferred using segmentation cues and temporal constraints. To obtain the action representation, estimated joints are grouped into five body parts: Head, Left arm, Right arm, Left leg, and Right leg (see Fig. 9.8). Efficient Contrast mining algorithm is used to gather distinctive co-occurring spatial configuration of body parts, called spatial-part-sets, and co-occurring pose sequences of body parts in temporal domain, called temporal-part-sets. Similarly, for test videos, part-sets are detected using estimated poses and represented using a histogram of part-sets. This histogram is then classified using a Support Vector Machine (SVM). Representing actions in terms of poses and part-sets gives a better visual interpretation and is compact as compared to high-dimensional, low-level representations. It is also robust to variations in body part movements since it can model the temporal movements effectively.

Shape Models

A joint shape-motion descriptor is proposed in [26], which combines shape features from an appearance model and motion features from optical flow field to capture distinct properties of an action. The approach represents an action as a sequence of prototypes for flexible action matching in video sequences. In training, action interest regions are localized and shape-motion descriptors are extracted. Using hierarchical K-means clustering, an action prototype tree is learned in a joint shape and motion space. The training sequences are represented as a labeled prototype sequence. In testing, humans are detected and tracked using appearance information, while frame-to-prototype correspondence is established by maximizing joint probability of the actor location and action prototype by searching the learned prototype tree. Finally, the action is recognized using dynamic prototype sequence matching.

The shape descriptor for an action interest region is represented as a feature vector by dividing the region of interest into a square grid. Shape observations are gathered using background subtraction or from appearance likelihood maps. For the appearance likelihood map, an appearance model is built and is used to assign a probability to each pixel in the specified region. An accumulation of such probabilites

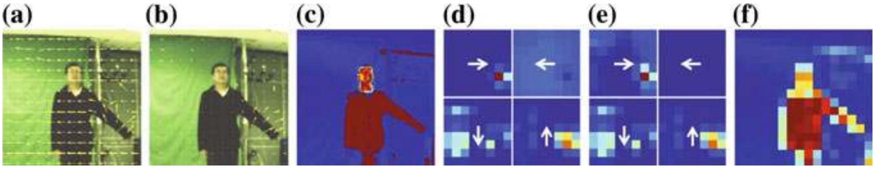

Fig. 9.9 Computing shape-motion descriptor [26]: **a** Raw optical flow field. **b** Motion-compensated optical flow field. **c** Appearance likelihood map. **d** Motion descriptor from the raw optical flow field. **e** Motion descriptor from the motion-compensated optical flow field. **f** Shape descriptor

for each grid square is used as a shape feature vector. The feature vector is L_2 normalized to get the final shape descriptor, as seen in Fig. 9.9f.

The motion descriptor for an action interest region is represented as a feature vector of *Q*uantized, *B*lurred, *M*otion-compensated *F*low (*QBMF*). The motion flow feature is calculated similar to [14], where optical flow is computed and divided into horizontal and vertial components and then background motion is removed by subtracting the medians of flow fields to get median-compensated flow fields. Motion-compensated flow fields (see Fig. 9.9b) are half-wave rectified into four nonnegative channels and each one of them is blurred using a Gaussian Kernel to get motion observations. The four channels are L_2 normalized and concatenated to get the raw motion descriptor. Finally, the raw motion descriptor is L_2 normalized to get the final motion descriptor. Figure 9.9d, e, shows the four channels of the motion descriptor, where grid intensity indicates motion strength and the arrow indicates the dominant motion orientation at that grid.

The results demonstrate good action recognition performance under moving camera and dynamic background. The reason is that the approach models the correlation between shape and motion using action prototypes in a joint feature space, which allows the method to tolerate such complex conditions.

9.3.2 Dictionary Learning

Dictionary learning is an important step in forming the representation of actions. It is most commonly employed in a Bag-of-Words (BoW) framework by using either low-level features such as STIP [32, 33] or high-level features such as human poses [76]. Typically, an unsupervised learning technique such as K-means is applied to cluster local features, and then the features from each video sequence are mapped to these cluster centers to get a histogram representation.

Sparse coding is also very popular and has been successfully applied to many computer vision problems such as image classification [84], visual saliency [83], and image restoration [55]. Sparse coding attempts to approximate an input sample using a linear combination of a few items from an overcomplete dictionary. Dictionary learning methods can be categorized into unsupervised and supervised learning.

The dictionary learning is unsupervised when the goal is to minimize reconstruction error of the original samples in order to build a dictionary. Supervised learning techniques can form a category specific dictionary that promotes discrimination between classes; this discriminatory term is added to the objective function which is to be optimized.

In the following sections, we will explore new techniques for dictionary learning that have produced notable results on UCF Sports dataset.

9.3.2.1 Multi-Task Sparse Learning

Multi-Task Sparse Learning (MTSL) [89] aims to construct a given test sample with multiple features and very few bases. In this framework, each feature modality is considered as a single task in MTSL to learn a sparse representation. These tasks are generated from multiple features extracted from the same visual input (video), hence they are interrelated. A Beta Process (BP) prior is used to learn a compact dictionary and infer sparse structure shared across all tasks.

Each video is represented using two types of features: (1) a histogram and (2) co-occurrence features. Histogram representation has achieved state-of-the-art performance and is frequently used in a bag-of-words framework. However, it ignores the geometric distribution of local features. Co-occurrence feature makes use of spatio-temporal proximity distribution of local features in a video to utilize geometric context.

A task is generated from each modality of features coming from the same video sequences. Each individual task is defined as learning a sparse representation of the video in one feature space. Multi-task learning improves the performance of each individual task by sharing information between related tasks. A target sample $y_j, j = 1, \ldots, J$ with J tasks, obtained from a video is represented in terms of a dictionary D_j, a sparse coefficient θ_j and a residual error ε_j. A sparse coefficient vector is the product of a binary vector z_j, defining which dictionary terms are used to represent a sample, and a weight vector ω_j. A Beta Process is used to formulate the dictionary as well as the binary vector. A graphical model based representation of MTSL is shown in Fig. 9.10. The bottom layer consists of individual models with task-related parameters. In the middle layer, tasks are connected via common prior of the tasks and the top layer is the hyperprior invoked on the parameters of the prior. In the model, the variables are infered given the training samples. Gibbs sampling is used to update them iteratively. All variables except the dictionary D are initialized randomly. However, the dictionary is initialized using K-SVD [53]. K-SVD learns an overcomplete dictionary for sparse representation. Once initialized, MTSL obtains a compact and discriminative dictionary by Gibbs sampling. To classify a test video, the sparse representation is obtained by MTSL model using the learned dictionary. It is then classified using the reconstruction error accumulated over all the tasks.

This method classifies UCF Sports actions using a Leave-One-Out setup and achieves an accuracy of 92.67 %. MTSL approach combined multiple features efficiently to improve the recognition performance. Its robust sparse coding technique

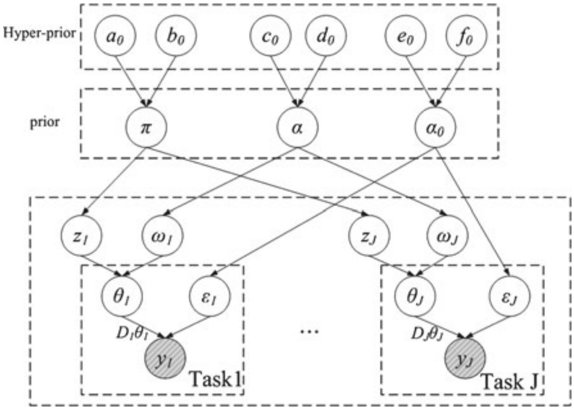

Fig. 9.10 Hierarchical Bayesian model representation of Multi-Task Sparse Learning (MTSL) approach [89]. Details of the parameters can be found in the paper

mines correlations between different tasks to obtain a shared sparsity pattern which is ignored if each task is learned individually. By using the Beta Process, the reconstruction coefficient vector is sparse, and it is distinct from the widely used l_1 regularized sparseness, which has many small coefficient values, but not exactly zero.

9.3.2.2 Label Consistent K-SVD

To learn a discriminative dictionary for sparse coding, a label consistent K-SVD (LC-KSVD) algorithm is proposed in [28]. During the dictionary learning process, the approach uses class label information of training data and associates label information with each dictionary item to enforce discriminability in sparse codes.

The above method presents a supervised learning algorithm to learn a reconstructive and discriminative dictionary for sparse coding. The objective function has a reconstruction error term and has two formulations for sparsity constraints: (1) L_0 norm and (2) L_1 norm. It also introduces a new label consistency constraint called "discriminative sparse code error." This term encourages the features from the same class to have similar sparse codes and those that belong to different classes have different sparse codes. Each item in the dictionary is chosen in a way that it represents a subset of training features that are ideally from a single class, so that each dictionary item can be associated to a particular label. This makes the correspondence between dictionary items and the class labels. The last term in the objective function is the classification error term. This term is a linear predictive classifier, and enables joint dictionary and classifier construction. The objective function is efficient and achieves a good classification performance; it also allows feature sharing among classes. By including the classification term, the objective function enforces a label consistency constraint on the sparse code with respect to the dictionary. It also makes the learned dictionary adaptive to underlying stucture of the training data. Learning the dictionary and classifier separately might make the dictionary suboptimal.

Two different approaches are presented for dictionary learning: (1) LC-KSVD1 (2) LC-KSVD2. The objective function for LC-KSVD1 uses the reconstruction error term and the label consistency regularization term. LC-KSVD2 has similar objective function with a classification term added to it. This helps the algorithm jointly learn a single overcomplete dictionary and a linear classifier. The parameters are initialized by several iterations of K-SVD and multiple ridge regression model. The dictionary is constructed by minimizing the error terms and satisfying the sparsity constraints.

LC-KSVD accesses the entire training set at every iteration to optimize the objective function. This can be difficult in a situation with limited memory resources. Therefore, an incremental dictionary learning algorithm is presented that can employ the same learning framework, but with limited memory resources.

This approach is evaluated using a Leave-One-Out experimental setup as well as five-fold cross-validation and it achieves the accuracies of 95.7 and 91.2 %, respectively. The results show that the method performs better than other sparse coding techniques that learn the dictionary and then use a one-against-all classifier to obtain classification results. Instead of iteratively solving subproblems to get a global solution, the proposed method is able to simultaneously learn the dictionary, discriminative sparse coding parameters, and the classifier parameters.

9.3.3 Action Classification

After passing through the stages of feature extraction and dictionary learning, the next step is to form a video representation. Features studied in Sect. 9.3.1 are generally used in the popular bag-of-words (BoW) [16, 77, 78] framework, where local features are extracted from video sequences and mapped to a prelearned dictionary. The dictionary is a collection of code words obtained by clustering features, generally by K-means. After assigning each feature to a specific code word in a dictionary, the video is represented using a histogram. This type of representation is simple and efficient. Histogram representations are relatively robust to occlusion and viewpoint changes.

Once a video representation is obtained, the final step is to classify the actions. The technique for classification can either be generative, e.g., HMM [1, 19, 82], or discriminative, e.g., CRF [44, 63]. It can also be as simple as a Nearest Neighbor classifier or more complex methods such as mapping to a Grassmann manifold [23, 51]. Over the years, a variety of techniques have been proposed for performing action classification on UCF Sports. Some of these methods are: template-based [57, 58], hough-based [10, 21, 86], manifold learning [23], randomized trees [51], multiple kernel learning [81], pattern mining [79], and several others. The following sections present some of the notable techniques that show superior classification results on UCF Sports.

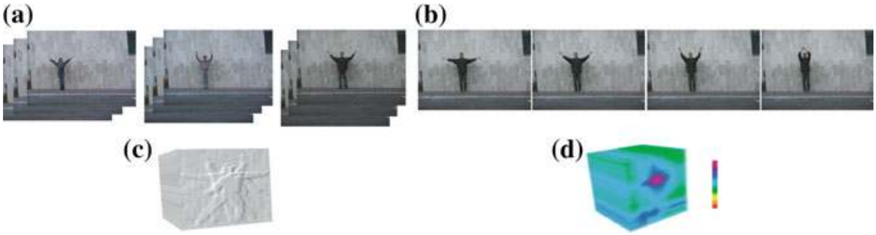

Fig. 9.11 Action MACH [57]: **a** Jumping Jack action from Weizmann action dataset. **b** 3D action MACH filter for Jumping Jack action. **c** Wave2 action from Weizmann action dataset. **d** Normalized correlation response for the Wave2 action using action MACH filter

9.3.3.1 Template-Based Method

Template-based matching emerged as an early solution for classifying actions. *Action Maximum Average Correlation Height (MACH)* filter proposed in [57] applies template-based matching to 3D spatio-temporal volume (video) having vector-valued data at each pixel location. The filter is generic and embeds intraclass variability into a single template. This capability enables the method to effectively discriminate a wide range of actions on UCF Sports dataset.

A *MACH* filter uses the training instances of a class to learn a single template by optimizing four performance metrics: Average Correlation Height (ACH), Average Correlation Energy (ACE), Average Similarity Measure (ASM), and Output Noise Variance (ONV). This gives a two-dimensional template which, when correlated with a testing image using a FFT transform in frequency domain, results in a response giving the most likely location of the object. The approach used in *Action MACH* extends this filter to be applied to spatio-temporal volumes and be able to fully utilize the temporal information.

During training, derivatives of spatio-temporal volumes, obtained from training video sequences, are represented in frequency domain by performing a 3D FFT transform. This 3D matrix is reshaped into a long column vector and is synthesized by minimizing average correlation energy, average similarity measure, output noise variance, and maximizing the average correlation height. After obtaining the 1D vector, an inverse 3D Fourier transform gives the *Action MACH* filter. However, this filter is only good for scalar values at each pixel location. To use vector-valued data, Clifford Fourier transform is applied, which is a generalization of the traditional scalar-valued Fourier transform.

Motion estimation in the video sequences is performed using Spatio-Temporal Regularity Flow (SPREF) [2]. SPREF computes the directions along which the sum of the gradients of a video sequence is minimized. Hence, a 3D vector field is generated having three values at each location that represent the direction along which the intensities of pixels change the least. This vector-valued data is then used to get the final *Action MACH* filter by applying Clifford Fourier transform (see Fig. 9.11a, b).

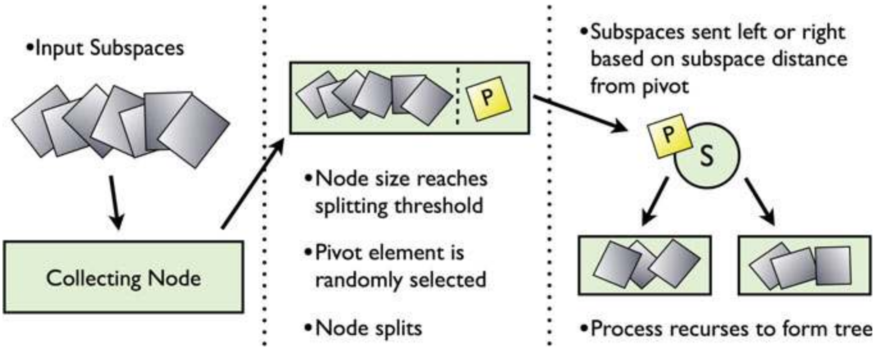
•Input Subspaces
Collecting Node
P
•Node size reaches splitting threshold
•Pivot element is randomly selected
•Node splits
•Subspaces sent left or right based on subspace distance from pivot
P
S
•Process recurses to form tree

Fig. 9.12 Subspace tree construction [51]. *Left* Subspaces are added to a node. *Middle* If the node size exceeds a threshold it is split. *Right* The subspaces are split and assigned to *right* and *left* child node

To recognize an action in a test video sequence, the *Action MACH* filter is applied everywhere in the spatio-temporal volume (see Fig. 9.11c, d). The peak value in the response of the filter is compared to a threshold value and if it is greater, the video is classified to be of that particular action class. This approach was the first one to be applied on UCF Sports dataset and gave encouraging results.

9.3.3.2 Subspace Forest

A novel structure called Subspace Forest is introduced in [51], which proposes a randomized forest based approximate nearest neighbor (ANN) method for subspaces. This stucture is applied to action recognition, where actions are represented as subspaces on a Grassmann manifold. A Subspace tree is constructed during training, and, at testing stage, an ANN-based approach is used to classify each video sequence.

An action from a video is represented as a sequence of thumbnail-sized images called *tracklets*. Each *tracklet* is a three-dimensional cube with height, width, and frame as its axes. All tracklets are designed to be of an equal size. Every *tracklet* is mapped as one point on the Grassmann manifold. The Grassmann manifold *Gr(r,n)* can be described as a set of *r*-dimensional linear subspaces of the *n*-dimensional vector space *V*. Points on the Grassmann manifold are subspaces and can be identified using orthogonal matrices. To map a *tracklet* on Grassmann manifold, the method first unfolds the *tracklet* along one of its axes and then applies QR factorization to get its orthogonal matrix. In this way, each *tracklet* is mapped onto the Grassmann manifold and represented by three subspaces coming from three tensor unfoldings. The distance between two tracklets is computed using the chordal distance by applying the component-wise sine function to the principal angles between subspaces.

A Subspace Tree (SSTree) is defined as a tree structure to sort points on Grassmann manifold. In this approach *tracklets*, obtained from video sequences, are sorted using their orthogonal basis. The SSTree is constructed by adding samples to an initially empty node, until the number of samples within a node become large enough to consider splitting (see Fig. 9.12). One of the elements of the node is selected as the pivot and chordal distance is computed between each element and the pivot. Based on the distance measure, i.e., greater or less than a threshold, the elements are added to the right and left child nodes, respectively. This recursive process forms the tree and all the samples are trickled down to the leaf nodes. The algorithm uses two different variations for splitting a node: (1) Median spliting (2) Entropy splitting. In median splitting, the pivot is selected randomly and the splitting threshold is selected by computing the median chordal distance between the pivot and all the remaining elements of the node. In entropy splitting, the pivot is also chosen randomly and the splitting threshold is selected using entropy over normalized histogram of distances computed between pivot and remaining elements. If the entropy falls below a threshold, then the distances are split into two clusters and the midpoint between cluster centers is used as a splitting threshold.

A variation of SSTree is also presented in an approach called Random Axes SSTree (RA-SSTree). In this tree structure, every new child node randomly selects an unfolding axes.

A Subspace Forest is defined as a collection of SSTrees. The forest size is chosen as a multiple of three for SSTree, so that all three unfolding axes of the *tracklet* have an equal number of SSTrees. In case of RA-SSTree, any number can be chosen.

To recognize the action class of a given video, first its *tracklet* is computed and then unfolded along X, Y, and T dimension to get three orthogonal matrices. All the orthogonal matrices are presented to the corresponding subspace trees in the forest to find out the approximate nearest neighbors (ANN). The final classification is done by the label of the K-Nearest-Neighbors (KNN) from each leaf node using a majority voting scheme.

Using subspace forest on Grassmann manifold, the method is able to achieve superior results as compared to *Action MACH*. The method has the ability to scale to larger real-world problems. It is also very simple to implement and has less parameters as compared to bag-of-features framework.

9.4 Action Localization in Sports Videos

The task of localization involves identifying the spatio-temporal volume in which the action is taking place. The simplest way to do this is to learn an action recognition classifier and use a sliding window approach. This method slides in the spatio-temporal volume of the video to select the subvolume with the maximum classifier score. By finding the volume with the highest score, a bounding box is placed at each frame to find the overlap with the ground truth. The label of the volume as well as the average percentage overlap (between ground truth and predicted bounding box) over all frames is used as a measure to judge whether the action was correctly localized.

Action *recognition approaches* that use a high-level representation, such as space-time shape models [22], silhouettes [74], human poses [76] or motion history images [7], have the potential ability to localize an action. However, this is not typically feasible in case of global action representations, e.g., bag-of-words histogram, which lose the spatial information.

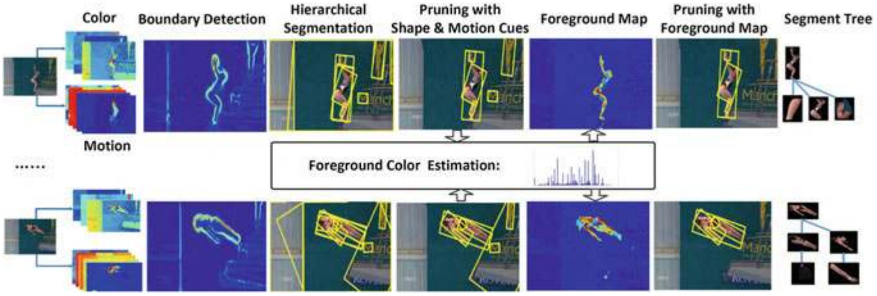
Color
Boundary Detection
Hierarchical Segmentation
Pruning with Shape & Motion Cues
Foreground Map
Pruning with Foreground Map
Segment Tree
Motion
......
Foreground Color Estimation:

Fig. 9.13 Process to extract hierarchical space-time segments [40]. Each video is used to compute a boundary map using color and motion cues. The map is used to extract hierarchical segments. Irrelevant segments are pruned, firstly using shape and motion cues, then using foreground map to obtain a segment tree

Some of the recent methods achieving best localization performances are elaborated in the following sections.

9.4.1 Hierarchical Space-Time Segments

Representing an action by Hierarchical Space-Time Segments [40] has shown that preserving the static and nonstatic space-time segments along with their hierarchical relationships yields good action localization results. This approach uses a two-level hierarchiy: first level consists of root space-time segments and second level has parts of the root. Without utilizing bounding box annotations to train any body or body part detector, the algorithm uses an unsupervised approach to extract hierarchical space-time segments. Each step of the algorithm is elaborated below.

First, using color and motion information, a frame segmentation method is designed that preserves segments of the body and its parts, while suppressing the background. A boundary map is computed using color and motion channels to be utilized in forming an Ultrametric Contour Map (UCM). UCM gives a hierarchical segmentation of a video frame. The segment tree is traversed to prune irrelevant parts using motion and shape cues. For further pruning, a foreground map is built based on structure and global color cue over the whole video sequence, yielding a set of candidate segment trees, (see Fig. 9.13). In the remaining segment trees, each segment is tracked forward and backward in the video to get space-time segments. These space-time segments are used to train a Bag-of-Words framework with linear SVM. In testing, space-time segments are identified with a positive contribution to the video classification (see Fig. 9.14).

Thus, by using static and nonstatic parts, the method is able to achieve good classification and localization performance. The static information helps in extracting body parts that are not necessarily in motion, hence resulting in better localization. Sample results can be seen in Fig. 9.15. The method reports an accuracy of 42.1 % measured as average Intersection-Over-Union (IOU) over a subset of frames.

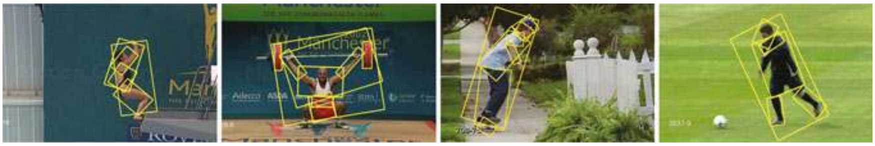

Fig. 9.14 Extracted segments from video frames [40]. Segments are outlined by *yellow* boxes. Boxes within a box show child–parent relationship

9.4.2 Spatio-Temporal Deformable Part Models

A natural extension of Deformable Part Models from 2D to a 3D for action localization is given by Spatio-Temporal Deformable Part Models (SDPM) [69]. In this method, a separate action model is learned for each class by selecting the most discriminative 3D subvolumes as parts and establishing spatio-temporal relations between them. The deformity in spatio-temporal volume that this approach yields empowers capturing the intraclass variabilities and becoming robust to background clutter.

The model consists of a root filter and many part models. Every part is defined by its part filter, anchor position, and coefficients of deformation cost. In the training stage, positive instances are selected from a single box of one cycle of an action. Negative instances are selected from positive volumes of other action classes, as well as by randomly drawing volumes from the background at multiple scales. HOG3D [29] features are extracted and a SVM is trained accordingly. Similarly, for training part models, HOG3D features are extracted at twice the resolution enabling them to capture more detail. Once SVM is applied, subvolumes having higher weights (i.e., more discriminative) are selected as parts, while others are ignored. After the initial model is obtained, latent SVM is used to update the model, while treating position of *i*th part as a latent variable (see Fig. 9.16)

In the testing stage, a spatio-temporal feature pyramid is built using HOG3D features at different spatial scales for the query video. A template-based sliding window approach is applied in 3D volume and the placement with the highest score is chosen to be the location of the action. This placement is defined by the location of the root and part filters (see Fig. 9.17).

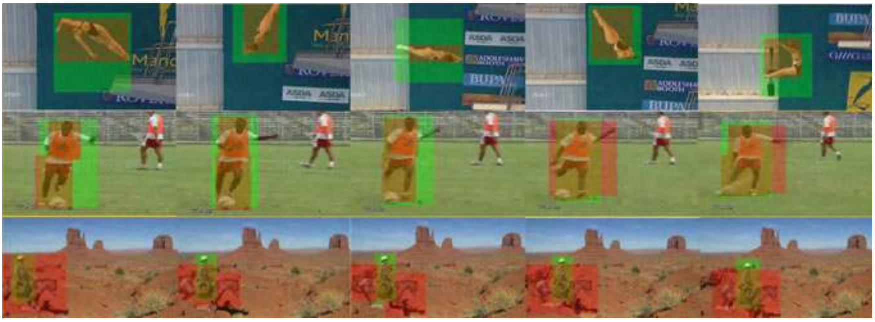

Fig. 9.15 Action localization results [40]. *Green* area denotes the ground truth annotation, whereas the *red* area shows the localization result

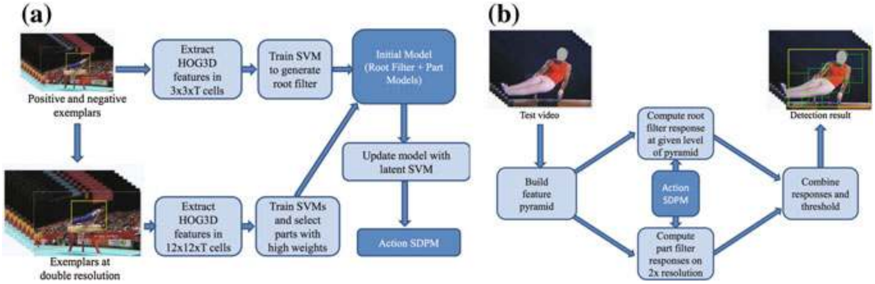
(a)
Positive and negative exemplars
Extract HOG3D features in 3x3xT cells
Train SVM to generate root filter
Initial Model (Root Filter + Part Models)
Update model with latent SVM
Action SDPM
Exemplars at double resolution
Extract HOG3D features in 12x12xT cells
Train SVMs and select parts with high weights
(b)
Test video
Build feature pyramid
Compute root filter response at given level of pyramid
Action SDPM
Compute part filter responses on 2x resolution
Combine responses and threshold
Detection result

Fig. 9.16 Spatio-Temporal Deformable Part Models (STPM) [69]. **a** Training process shows the extraction of HOG3D features and learning the model using latent SVM. **b** Testing process with the final result showing the root (*yellow*) and its parts (*green*)

This approach has shown to be effective as the parts exclude most of the background giving better localization and focus on distinctive locations within an action. This method is different from other approaches as it explicitly models the intraclass variability using part deformations, and by using global and part templates, it is able to find the best location for an action in the scale, space, and time. The method achieves state-of-the-art results on UCF Sports dataset, as shown in Fig. 9.18.

9.5 Discussion

The earlier action recognition benchmarks [6, 59] were recorded in a controlled setting having static cameras with static and uncluttered backgrounds. The actions were performed by a few selected actors and appeared without any occlusions. This was improved by changing the source of videos to television broadcast and movies in datasets such as UCF Sports [34, 41, 47, 57, 70]. Action videos from these types of

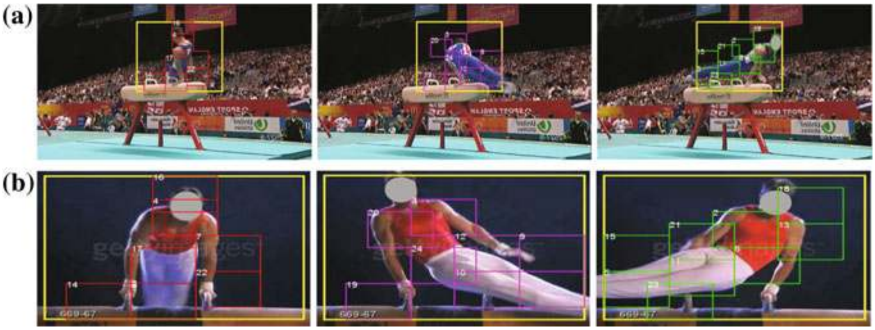

Fig. 9.17 SDPM [69] localization. **a** Root and part filter locations in a training frame for *Swing-Bench* action. **b** Localization result in a test video. Root (*yellow*) and part (*red*, *magenta*, and *green*) locations are shown for both train and test examples. Each column shows the middle frame of a three temporal stage model

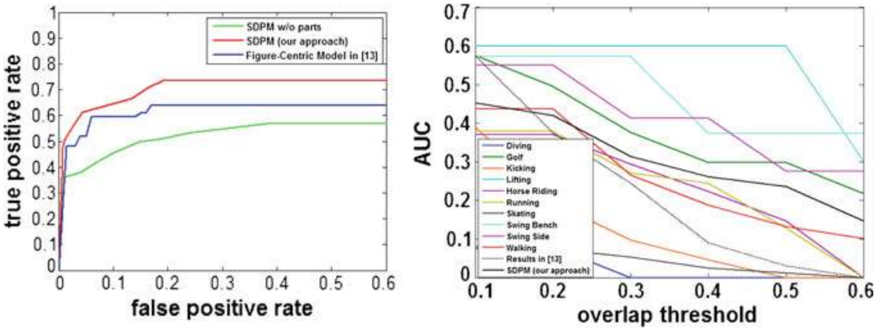

Fig. 9.18 SDPM [69] results. *Left* ROC at 0.2 overlap threshold; compared with figure-centric model [34]. *Right* Area Under Curve (AUC) for overlap threshold ranging from 0.1 to 0.6

sources presented a diversity of appearances, occlusion, and varying backgrounds. Although these videos were recorded in an unconstrained environment, they were produced by professional crew in favorable conditions, e.g., using selected viewpoints. Later, the focus shifted toward collecting videos from online repositories, e.g., YouTube. Videos collected from such sources were named to be videos "in the wild" [36, 64]. Since the videos are uploaded by a diverse group of users, they present a wide variety of challenges in a totally unconstrained environment. These datasets have been utilized for both localizing and recognizing actions. In the rest of this section, we discuss some of the common and crucial shortcomings in many of the existing action recognition and localization methods which are expected to be addressed in the future techniques.

Evaluating on temporally trimmed videos: even though the benchmarks have evolved to be more challenging by being recorded in an unconstrained setting and having a large number of action classes, the majority of the existing action datasets suffer from a crucial shortcoming: the collected videos are carefully trimmed to only contain the action of interest. Hence, the focus of the current action recognition methods has been toward classifying actions in temporally trimmed videos which is an unrealistic assumption. In addition, in many of the existing datasets, the action is performed by only a single person, as compared to a group of people. This makes these datasets even further simplistic and unrealistic compared to analyzing human actions in a natural setting. In videos having a single actor, the main task is to identify and recognize the motion of the actor, while separating it from background clutter. However, realistic scenarios would have several actions being performed simultaneously by different actors with massive inter- and intraclass variability. Therefore, the next generation of action recognition and localization methods are expected to address these two major shortcomings and be able to perform their task on temporally untrimmed videos [27] with potentially multiple actors performing various actions.

The task of recognizing multiple actions simultaneously will introduce new challenges to be explored such as: co-occurrence of actions, action-to-action occlusion, and interclass dependencies. Potential applications which would require localization of multiple actions include: video surveillance, automatic understanding of sports videos, or crowd analysis. For instance, automatic video surveillance requires detecting (multiple) actions in real time, so an unwanted event can be predicted and prevented. Such a system can also highlight the level of security threat of one action over another, and therefore prioritize the localization of such actions.

Performing a forced-choice classification: thus far, action recognition has been defined as a forced-choice classification task, which means a video has to belong to one of the predefined action classes. Consequently, the majority of existing action recognition methods have a poor performance when dealing with an unseen action class or a clip which simply does not contain any particular action. A potential alternative way of understanding actions is to *describe* them instead of classifying them. Even though there exists an extremely wide variety of actions in the real world, many of them share vast similarities in an atomic level. For example, the action of Pole Vault can be broken down into running, jumping, landing, and then standing up. Therefore, it is often feasible to describe an action using a universal lexicon of lower level actions, sometimes called action attributes [17, 37]. Hence, it is a worthwhile effort for the future action recognition techniques to understand the basic elements of human actions and devise a simple and comprehensive *description* for an action rather than a forced-choice classification.

Employing Exhaustive search as the search strategy: recently, several action localization methods that employ mid-to-high level representations [31, 69] which can effectively model the spatio-temporal structure of an action have been proposed. However, many of these approaches perform an exhaustive search using a sliding window, in temporal, spatial, or spatio-temporal domain, to find the desired location of the action. This approach is particularly inefficient as all possible spatio-temporal locations over different scales and aspect ratios have to be evaluated. Recently, efficient search strategies, such as selective search or object proposal [3, 9, 13, 15, 72], were shown to be more efficient than sliding window for object detection. Potentially, action localization methods can also adopt a similar approach [25] and utilize similar search strategies in order to increase the efficiency of their search in the spatio-temporal domain.

9.6 Conclusion

In this chapter, we overviewed the prominent action localization and recognition methods for sports videos. We adopted UCF Sports as the benchmark for evaluating the discussed techniques, as it includes a wide range of unconstrained videos categorized into 10 different sports collected from broadcast television channels. We provided an overview of the characteristics of UCF Sports as well as detailed statistics of the techniques evaluated on this dataset along with the evolution of their performance over time. To provide further technical details, we decomposed action recognition into three major steps of feature extraction, forming the video representation using dictionary learning, and classification. For each step, we studied the approaches which yield superior results on sports videos and discussed the reasons behind their success. Similarly, we presented the challenges faced in action localization, elaborated the reasons behind its intricacy, and overviewed several recent methods for this task, which have achieved promising results on sports videos. Lastly, we presented a number of insights acquired from summarizing the discussed action recognition and localization methods. We argued that conducting the recognition on temporally untrimmed videos and attempting to describe an action, instead of performing a forced-choice classification, are crucial for analyzing the human actions in a pragmatic environment.

References

1. Ahmad M, Lee SW (2008) Human action recognition using shape and CLG-motion flow from multi-view image sequences. Pattern Recognit 41(7):2237–2252
2. Alatas O, Yan P, Shah M (2007) Spatio-temporal regularity flow (SPREF): its estimation and applications. IEEE Trans Circuits Syst Video Technol 17(5):584–589
3. Alexe B, Heess N, Teh Y, Ferrari V (2012) Searching for objects driven by context. In: Neural information processing systems (NIPS)
4. Ali S, Shah M (2010) Human action recognition in videos using kinematic features and multiple instance learning. IEEE Trans Pattern Anal Mach Intell (TPAMI) 32(2):288–303
5. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell (TPAMI) 24(4):509–522
6. Blank M, Gorelick L, Shechtman E, Irani M, Basri R (2005) Actions as space-time shapes. In: Computer vision and pattern recognition (CVPR)
7. Bobick AF, Davis JW (2001) The recognition of human movement using temporal templates. IEEE Trans Pattern Anal Mach Intell (TPAMI) 23(3):257–267
8. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell (TPAMI) 6:679–698
9. Carreira J, Sminchisescu C (2010) Constrained parametric min-cuts for automatic object segmentation. In: Computer vision and pattern recognition (CVPR)
10. Cheng SC, Cheng KY, Chen YPP (2013) GHT-based associative memory learning and its application to human action detection and classification. Pattern Recognit 46(11):3117–3128
11. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: Computer vision and pattern recognition (CVPR)
12. Dalal N, Triggs B, Schmid C (2006) Human detection using oriented histograms of flow and appearance. In: European conference on computer vision (ECCV)
13. Dollar P (2010) A seismic shift in object detection. http://pdollar.wordpress.com/2013/12/10/a-seismic-shift-in-object-detection
14. Efros A, Berg A, Mori G, Malik J (2003) Recognizing action at a distance. In: International conference on computer vision (ICCV)
15. Endres I, Hoiem D (2014) Category-independent object proposals with diverse ranking. IEEE Trans Pattern Anal Mach Intell (TPAMI) 36:222–234
16. Everts I, van Gemert J, Gevers T (2013) Evaluation of color stips for human action recognition. In: Computer vision and pattern recognition (CVPR)
17. Farhadi A, Endres I, Hoiem D, Forsyth D (2009) Describing objects by their attributes. In: computer vision and pattern recognition (CVPR)
18. Fei-Fei L, Perona P (2005) A Bayesian hierarchical model for learning natural scene categories. In: Comput vision and pattern recognition (CVPR), vol 25, pp 24–531
19. Feng X, Perona P (2002) Human action recognition by sequence of movelet codewords. In: International symposium on 3D data processing, visualization, and transmission. IEEE, pp 717–721

20. Freeman WT, Adelson EH (1991) The design and use of steerable filters. IEEE Trans Pattern Anal Mach Intell (TPAMI) 13(9):891–906
21. Gall J, Yao A, Razavi N, Van Gool L, Lempitsky V (2011) Hough forests for object detection, tracking, and action recognition. IEEE Trans Pattern Anal Mach Intell (TPAMI) 33(11):2188–2202
22. Gorelick L, Blank M, Shechtman E, Irani M, Basri R (2007) Actions as space-time shapes. IEEE Trans Pattern Anal Mach Intell (TPAMI) 29(12):2247–2253
23. Harandi MT, Sanderson C, Shirazi S, Lovell BC (2013) Kernel analysis on Grassmann manifolds for action recognition. Pattern Recognit Lett 34(15):1906–1915
24. Harris C, Stephens M (1988) A combined corner and edge detector. In: Alvey vision conference, vol 15. Manchester, p 50
25. Jain M, van Gemert JC, Bouthemy P, Jégou H, Snoek C (2014) Action localization by tubelets from motion. In: Computer vision and pattern recognition (CVPR)
26. Jiang Z, Lin Z, Davis LS (2012) Recognizing human actions by learning and matching shape-motion prototype trees. IEEE Trans Pattern Anal Mach Intell (TPAMI) 34(3):533–547
27. Jiang YG, Liu J, Zamir AR, Laptev I, Piccardi M, Shah M, Sukthankar R (2014) Thumos challenge: action recognition with a large number of classes
28. Jiang Z, Lin Z, Davis L (2013) Label consistent K-SVD—learning a discriminative dictionary for recognition
29. Kläser A, Marszalek M, Schmid C (2008) A spatio-temporal descriptor based on 3d-gradients. In: British machine vision conference (BMVC)
30. Kovashka A, Grauman K (2010) Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: Computer vision and pattern recognition (CVPR)
31. Lan T, Wang Y, Mori G (2011) Discriminative figure-centric models for joint action localization and recognition. In: International conference on computer vision (ICCV)
32. Laptev I (2005) On space-time interest points. Int J Comput Vis 64(2–3):107–123
33. Laptev I, Lindeberg T (2003) Space-time interest points. In: International conference on computer vision (ICCV)
34. Laptev I, Marszalek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: Computer vision and pattern recognition (CVPR)
35. Le Q, Zou W, Yeung S, Ng A (2011) Learning hierarchical invariant spatiotemporal features for action recognition with independent subspace analysis. In: Computer vision and pattern recognition (CVPR)
36. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos "in the wild". In: Computer vision and pattern recognition (CVPR)
37. Liu J, Kuipers B, Savarese S (2011) Recognizing human actions by attributes. In: Computer vision and pattern recognition (CVPR)
38. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
39. Lucas B.D, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: International joint conference on artificial intelligence (IJCAI)
40. Ma S, Zhang J, Cinbis N, Sclaroff S (2013) Action recognition and localization by hierarchical space-time segments. In: International conference on computer vision (ICCV)
41. Marszalek M, Laptev I, Schmid C (2009) Actions in context. In: Computer vision and pattern recognition (CVPR)
42. Matas J, Chum O, Urban M, Pajdla T (2002) Robust wide baseline stereo from maximally stable extremal regions. In: British machine vision conference (BMVC)
43. Matikainen P, Hebert M, Sukthankar R (2009) Action recognition through the motion analysis of tracked features. In: ICCV workshops on video-oriented object and event classification
44. Mendoza M.Á, De La Blanca NP (2008) Applying space state models in human action recognition: a comparative study. In: International Workshop on Articulated Motion and Deformable Objects. Springer, pp 53–62

45. Messing R, Pal C, Kautz H (2009) Activity recognition using the velocity histories of tracked keypoints. In: International conference on computer vision (ICCV)
46. Mikolajczyk K, Schmid C (2005) A performance evaluation of local descriptors. IEEE Trans Pattern Anal Mach Intell (TPAMI) 27(10):1615–1630
47. Mikolajczyk K, Uemura H (2008) Action recognition with motion-appearance vocabulary forest. In: Computer vision and pattern recognition (CVPR)
48. Mikolajczyk K, Zisserman A, Schmid C (2003) Shape recognition with edge-based features. In: British machine vision conference (BMVC)
49. Nelson RC, Selinger A (1998) Large-scale tests of a keyed, appearance-based 3-d object recognition system. Vis Res 38(15):2469–2488
50. Nowak E, Jurie F, Triggs B (2006) Sampling strategies for bag-of-features image classification. In: European conference on computer vision (ECCV), pp 490–503
51. O'Hara S, Draper B (2012) Scalable action recognition with a subspace forest. In: Computer vision and pattern recognition (CVPR)
52. Pope AR, Lowe DG (2000) Probabilistic models of appearance for 3-d object recognition. Int J Comput Vis 40(2):149–167
53. Qiu Q, Jiang Z, Chellappa R (2011) Sparse dictionary-based representation and recognition of action attributes. In: International conference on computer vision (ICCV)
54. Randen T, Husoy JH (1999) Filtering for texture classification: a comparative study. IEEE Trans Pattern Anal Mach Intell (TPAMI) 21(4):291–310
55. Ranzato M, Poultney C, Chopra S, LeCun Y (2006) Efficient learning of sparse representations with an energy-based model. In: Neural information processing systems (NIPS)
56. Raptis M, Kokkinos I, Soatto S (2012) Discovering discriminative action parts from mid-level video representations. In: Computer vision and pattern recognition (CVPR)
57. Rodriguez M, Ahmed J, Shah M (2008) Action Mach: a spatio-temporal maximum average correlation height filter for action recognition. In: Computer vision and pattern recognition (CVPR)
58. Sadanand S, Corso JJ (2012) Action bank: a high-level representation of activity in video. In: Computer vision and pattern recognition (CVPR)
59. Schuldt C, Laptev I, Caputo B (2004) Recognizing human actions: a local SVM approach. In: International conference on pattern recognition (ICPR)
60. Scovanner P, Ali S, Shah M (2007) A 3-dimensional sift descriptor and its application to action recognition. In: ACM international conference on multimedia
61. Shapovalova N, Raptis M, Sigal L, Mori G (2013) Action is in the eye of the beholder: eye-gaze driven model for spatio-temporal action localization. In: Neural information processing systems (NIPS)
62. Shi J, Tomasi C (1994) Good features to track. In: Computer vision and pattern recognition (CVPR)
63. Sminchisescu C, Kanaujia A, Metaxas D (2006) Conditional models for contextual human motion recognition. Comput Vis Image Underst 104(2):210–220
64. Soomro K, Zamir AR, Shah M (2012) Ucf101: A dataset of 101 human action classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012).
65. Sun J, Mu Y, Yan S, Cheong L (2010) Activity recognition using dense long-duration trajectories. In: International conference on multimedia and expo
66. Sun J, Wu X, Yan S, Cheong L, Chua T, Li J (2009) Hierarchical spatio-temporal context modeling for action recognition. In: Computer vision and pattern recognition (CVPR)
67. Tamrakar A, Ali S, Yu Q, Liu J, Javed O, Divakaran, A, Cheng H, Sawhney H (2012) Evaluation of low-level features and their combinations for complex event detection in open source videos. In: Computer vision and pattern recognition
68. Thi TH, Cheng L, Zhang J, Wang L, Satoh S (2012) Integrating local action elements for action analysis. Comput Vis Image Underst 116(3):378–395
69. Tian Y, Sukthankar R, Shah M (2013) Spatiotemporal deformable part models for action detection. In: Computer vision and pattern recognition (CVPR)

70. Tran D, Sorokin A (2008) Human activity recognition with metric learning. In: European conference on computer vision (ECCV)
71. Tran D, Yuan J (2012) Max-margin structured output regression for spatio-temporal action localization. In: Neural information processing systems (NIPS)
72. Uijlings J, van de Sande K, Gevers T, Smeulders A (2013) Selective search for object recognition. Int J Comput Vis 104(2):154–171
73. van Gool L, Moons T, Ungureanu D (1996) Affine/photometric invariants for planar intensity patterns. In: European conference on computer vision (ECCV)
74. Wang Y, Huang K, Tan T (2007) Human activity recognition based on r transform. In: Computer vision and pattern recognition (CVPR)
75. Wang H, Ullah MM, Kläser A, Laptev I, Schmid C (2009) Evaluation of local spatio-temporal features for action recognition. In: British machine vision conference (BMVC)
76. Wang C, Wang Y, Yuille A (2013) An approach to pose-based action recognition. In: Computer vision and pattern recognition (CVPR)
77. Wang H, Kläser A, Schmid C, Liu C (2011) Action recognition by dense trajectories. In: Computer vision and pattern recognition (CVPR)
78. Wang H, Kläser A, Schmid C, Liu CL (2013) Dense trajectories and motion boundary descriptors for action recognition. Int J Comput Vis 103(1):60–79
79. Wang L, Wang Y, Gao W (2011) Mining layered grammar rules for action recognition. Int J Comput Vis 93(2):162–182
80. Willems G, Tuytelaars T, van Gool L (2008) An efficient dense and scale-invariant spatio-temporal interest point detector. In: European conference on computer vision (ECCV)
81. Wu X, Xu D, Duan L, Luo J (2011) Action recognition using context and appearance distribution features. In: Computer vision and pattern recognition (CVPR)
82. Yamato J, Ohya J, Ishii K (1992) Recognizing human action in time-sequential images using hidden Markov model. In: Computer vision and pattern recognition (CVPR)
83. Yang J, Yang M (2012) Top-down visual saliency via joint CRF and dictionary learning. In: Computer vision and pattern recognition (CVPR)
84. Yang J, Yu K, Gong Y, Huang T (2009) Computer vision and pattern recognition (CVPR)
85. Yang Y, Ramanan D (2011) Articulated pose estimation with flexible mixtures-of-parts. In: Computer vision and pattern recognition (CVPR)
86. Yao A, Gall J, van Gool L (2010) A Hough transform-based voting framework for action recognition. In: Computer vision and pattern recognition (CVPR)
87. Yeffet L, Wolf L (2009) Local trinary patterns for human action recognition. In: International conference on computer vision (ICCV)
88. Yilmaz A, Shah M (2005) A novel action representation. In: Computer vision and pattern recognition (CVPR)
89. Yuan C, Hu W, Tian G, Yang S, Wang H (2013) Multi-task sparse learning with beta process prior for action recognition. In: Computer vision and pattern recognition (CVPR)