Unsupervised Action Discovery and Localization in Videos

Khurram Soomro, Mubarak Shah
Center for Research in Computer Vision (CRCV), University of Central Florida (UCF)
{ksoomro, shah}@cs.ucf.edu

Abstract

This paper is the first to address the problem of unsupervised action localization in videos. Given unlabeled data without bounding box annotations, we propose a novel approach that: 1) Discovers action class labels and 2) Spatio-temporally localizes actions in videos. It begins by computing local video features to apply spectral clustering on a set of unlabeled training videos. For each cluster of videos, an undirected graph is constructed to extract a dominant set, which are known for high internal homogeneity and in-homogeneity between vertices outside it. Next, a discriminative clustering approach is applied, by training a classifier for each cluster, to iteratively select videos from the non-dominant set and obtain complete video action classes. Once classes are discovered, training videos within each cluster are selected to perform automatic spatio-temporal annotations, by first over-segmenting videos in each discovered class into supervoxels and constructing a directed graph to apply a variant of knapsack problem with temporal constraints. Knapsack optimization jointly collects a subset of supervoxels, by enforcing the annotated action to be spatio-temporally connected and its volume to be the size of an actor. These annotations are used to train SVM action classifiers. During testing, actions are localized using a similar Knapsack approach, where supervoxels are grouped together and SVM, learned using videos from discovered action classes, is used to recognize these actions. We evaluate our approach on UCF-Sports, Sub-JHMDB, JHMDB, THUMOS13 and UCF101 datasets. Our experiments suggest that despite using no action class labels and no bounding box annotations, we are able to get competitive results to the state-of-the-art supervised methods.

1. Introduction

The problem of *action recognition* is to classify a video by assigning a label from a given set of annotated action classes, whereas in *action localization* the spatio-temporal extent of an action is detected and is also recognized. Existing action recognition and localization approaches [46, 18, 6] heavily rely on strong *supervision*, in the form of training videos, that have been manually collected, labeled and annotated. These approaches learn to *detect* an action using manually annotated bounding boxes and *recognize* using action class labels from training data. Since, *supervised* methods have the spatio-temporally annotated ground truth at their disposal, they can take advantage of learning detectors and classifiers by fine-tuning over the training data.

However, *supervised* algorithms have some disadvantages compared to *unsupervised* approaches, due to the difficulty of video annotation. First, a video may consist of several actions in complex cluttered background. Second, video level annotation in a *supervised* setting involves manually labeling the location (bounding box), the class of each action in videos and the temporal boundaries of each action, which is quite time con-

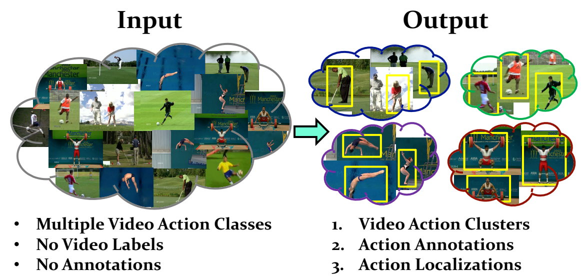

Figure 1. We tackle the problem of *Unsupervised Action Localization* without any action class labels or bounding box annotations, where a given collection of unlabeled videos contain multiple action classes. First, the proposed method discovers action classes by discriminative clustering using *dominant sets* (e.g. green and purple contours show clusters for *kicking* and *diving* actions, respectively) and then applies a variant of *knapsack* problem to determine spatio-temporal annotations of discovered actions (yellow bounding boxes). Then, these annotations and action classes are used together to train an action classifier and perform *Unsupervised Action Localization*.

suming. Third, actions vary spatio-temporally (i.e. in height, width, spatial location and temporal length) resulting in various tubelet deformations. Fourth, different people may have a different understanding of the temporal extent of an action, which results in unwanted biases and errors. *Collecting large amounts of accurately annotated action videos is very expensive for developing a supervised action localization approach, considering the growth of video datasets with large number of action classes [40, 12, 5, 44, 16, 1].* On the contrary, training an *unsupervised* system neither requires action class labels nor bounding box annotations. Given the abundance of unlabeled videos available on the Internet, *unsupervised* learning approaches provide a promising direction.

In this paper, we automatically discover action classes by discriminatively clustering a group of unlabeled training videos. Our approach begins by selecting a strongly coherent subset called a *dominant set* within each cluster, and trains a classifier for each action cluster to iteratively assign an action class to all the videos. Next, using these action classes, we propose a *Knapsack* approach to annotate actions in training videos. In this approach, we segment the video into supervoxels and using a combinatorial optimization framework we select the supervoxels that belong to the actor performing the action. Hence, we automatically obtain the ground truth: 1) action class labels and 2) actor bounding box annotations, for training videos and learn an action classifier to perform *Unsupervised Action Localization* (see Fig. 1).

In summary, this paper makes the following contributions: 1) Automatic discovery of action class labels using a new discriminative clustering approach with *dominant sets* (Sec. 3). 2) We propose a novel *Knapsack* approach with graph-based temporal constraints to annotate actions in training videos (Sec. 4). 3) The annotations within each cluster of videos are jointly selected by Binary Integer Quadratic Programming (BIQP) optimization to train action classifiers (Sec. 4.1). 4) Structural SVM is used to learn the pairwise relations of supervoxels within foreground action and foreground-background, which enforces that the supervoxels belonging to the action to be simultaneously selected (Sec. 5.1). 5) Lastly, we address a new problem of *Unsupervised Action Localization* (Sec. 5.2). In the next section we review existing literature relevant to our approach.

2. Related Work

Unsupervised Action Clustering aims to group videos of similar human actions into separate action classes, *without* any action localization. These approaches [33, 24, 51, 14, 15, 19] use local features to compute similarity among action videos. Wang et al. [48] use the coarse shape of human figures to match pairs of action images using a linear programming approach. Niebles et al. [24] use pLSA and LDA to learn intermediate topics associated with actions to cluster them. Yang et al. [51] discover sub-actions as motion primitives to construct a string matching similarity matrix for clustering. Jones and Shao [14] present a Feature Grouped Spectral Multigraph (FGSM) approach, that uses a spectral embedding on a feature graph to cluster actions. Liu *et al*. [19] suggest a hierarchical clustering multi-task learning method for jointly grouping and recognizing human actions. Jones and Shao [15] propose a Dual Assignment k-Means (DAKM) approach, which considers the contextual relations between actions and scenes for human action clustering. In contrast, we perform both action discovery as well as localization in an *unsupervised* manner. Our action discovery method employs a discriminatively-learned similarity as compared to standard low-level similarity metric (e.g. Euclidean), to iteratively cluster videos.

***Supervised* Action Localization** has been extensively studied in recent years [55, 54, 50, 8, 4, 17, 47, 10, 49, 29, 2, 20, 52, 34, 29, 45, 3, 22, 53, 41, 21, 57, 26, 56, 43, 55, 6, 37]. Since, we are the first to propose an *unsupervised* action localization method, this section only covers *supervised* works related to our approach. Among the supervoxel based methods, Jain *et al*. [9] proposed a hierarchical merging approach that produced multiple layers of segmentation for each video. Soomro *et al*. [36] use context walk with Conditional Random Field (CRF) to segment actions. These approaches use heuristics based on low-level feature similarity to define supervoxel merging criteria. They neither consider temporal connectedness nor spatial size of the actor within the action. Our *knapsack* approach is different in three key aspects: 1) it uses volume constraints to enforce detected action to be consistent with human spatial size, 2) temporal constraints to ensure that the detection is contiguous and well-connected, and 3) a discriminative selection criterion is learnt using Structured SVM to model supervoxel pairwise relations.

3. Action Discovery in Training Videos through Discriminative Clustering

In our proposed approach, we first aim to discover action classes from a set of unlabeled videos. We start by computing local feature similarity between videos to apply spectral clustering. Then, within each cluster, we construct an undirected graph to extract a *dominant set*. This subset is used to train a Support Vector Ma-

Algorithm 1 Algorithm to Discover K Action Classes

Input: Action Discovery Video Set $\mathcal{V}$
Output: Action Clusters $C_1 \dots C_K$

procedure DISCOVER ACTION CLASSES($\mathcal{V}$)
  $C_1 \dots C_K \Leftarrow spectral\_clustering(\mathcal{V})$
  $\Xi_1 \dots \Xi_K \Leftarrow dominant\_sets(C_1 \dots C_K)$
  $\mathbf{\Lambda} \Leftarrow \bigcup_{k=1}^{K} \widetilde{\Xi}_k$
  $C_1 \dots C_K \Leftarrow \Xi_1 \dots \Xi_K$
  do
    for $k = 1$ to K **do**
      $\mathbf{\Omega}_k \Leftarrow svm\_train(C_k, \bigcup_{k'=1,k'\neq k}^{K} C_{k'})$
      $C_k^{new} \Leftarrow select\_top(\mathbf{\Omega}_k, \mathbf{\Lambda}, \eta)$
      $C_k \Leftarrow C_k \cup C_k^{new}$
    end for
    $\mathbf{\Lambda} \Leftarrow \mathbf{\Lambda} \setminus \bigcup_{k=1}^{K} C_k^{new}$
  while $\mathbf{\Lambda} \neq \emptyset$
  return $C_1 \dots C_K$
end procedure

chine (SVM) classifier within each cluster and discriminatively selects videos from the *non-dominant set* to assign to one of the clusters in an iterative manner (see Alg. 1).

Let the index of unlabeled training videos range between $n = 1 \dots N$, where N is the total number of videos. Given this set of videos $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_N\}$, our goal is to discover video action classes. We initiate by obtaining a set of K clusters $C_1 \dots C_K$ using spectral clustering [23], where $C_k \subseteq \mathcal{V}, \forall k = 1 \dots K$. Since, the initial clusters can be noisy as they are computed using a low-level similarity metric (e.g. χ^2), we propose a data-driven approach to discriminatively refine each of these initial clusters. In this iterative approach, we select a subset $\Xi_k \subseteq C_k$, called *dominant set* [27, 28], from each cluster C_k. *Dominant set* clusters are known to maintain high internal homogeneity and in-homogeneity between items within the cluster and those outside it. For completeness, we present the basic definition and properties of *dominant set* next. For each cluster C_k we construct an undirected edge-weighted graph with no self-loops $\mathbf{G}_k(\mathbf{V}_k, \mathbf{E}_k, \omega_k)$, whose vertices correspond to videos, edges represent neighborhood relationships, weighting the video similarity (using C3D deep features [42]), and $\omega : E \rightarrow R_+^*$ is the (positive) weight function. The graph $\mathbf{G}_k$ is represented using a weighted adjacency (similarity) matrix, which is non-negative and symmetric $\mathbf{A}_k = a_k^{ij}$, where $a_k^{ij} = \omega_k(i,j)$ if $(i,j) \in \mathbf{E}_k$, and $a_k^{ij} = 0$ otherwise. As there are no self-loops in $\mathbf{G}_k$, the main diagonal of $\mathbf{A}_k$ is zero.

Let $\Xi_k \subseteq \mathbf{V}_k$ be a non-empty set, $i \in \Xi_k$ and $j \notin \Xi_k$, we define the function $\phi_k(i,j)$, which measures the relative similarity, using χ^2 similarity matrix, between vertices i and j with respect to the average similarity between vertex i and its neighbors in Ξ_k as $\phi_k(i,j) = a_k^{ij} - \frac{1}{|\Xi_k|}\sum_{i' \in \Xi_k} a_k^{ii'}$. For each vertex $i \in \Xi_k$ we recursively define its weight, with regard to Ξ_k, as follows:

$$\omega_{\Xi_k}(i) = \begin{cases} 1, & \text{if} |\Xi_k| = 1 \\ \sum_{j \in \Xi_k \setminus \{i\}} \phi_{\Xi_k \setminus \{i\}}(j,i)\,\omega_{\Xi_k \setminus \{i\}}(j), & \text{otherwise,} \end{cases} \quad (1)$$

and the total weight of Ξ_k is defined by $W(\Xi_k) = \sum_{i \in \Xi_k} \omega_{\Xi_k}(i)$. A non-empty subset of vertices $\Xi_k \subseteq \mathbf{V}_k$ such that $W(J) > 0$ for any non-empty $J \subseteq \Xi_k$, is said to be a dominant set if:

1. $\omega_{\Xi_k}(i) > 0$, for all $i \in \Xi_k$
2. $\omega_{\Xi_k \cup \{i\}}(i) < 0$, for all $i \notin \Xi_k$.

These *dominant sets* are obtained for each action cluster, C_k, using a continuous optimization technique known as *replicator dynamics* [27, 28], arising from evolutionary game theory. As shown in Algorithm 1, we group *non-dominant sets* into $\mathbf{\Lambda}$ and initialize clusters to *dominant sets*. Then, iteratively we train a one-vs-all linear SVM classifier $\mathbf{\Omega}_k$ for each cluster, using videos from the same cluster as positive examples and videos from the remaining clusters as negative examples. In each iteration, we test the classifier on $\mathbf{\Lambda}$ to select top η videos for each action and add them to their respective clusters, until the set $\mathbf{\Lambda}$ is empty.

4. Spatio-temporal Annotation of Training Videos using Knapsack

Given discovered action classes from our discriminative clustering approach, our aim is to annotate the action within each training video in every cluster. We begin by oversegmenting a video into supervoxels, where every supervoxel either belongs to the foreground action or the background. Our goal is to select a group of supervoxels that collectively represent an action. We achieve this goal by solving the *0-1 Knapsack problem: Given a set of items (supervoxels), each with a weight (volume of a supervoxel) and a value (score of a supervoxel belonging to an action), determine the subset of items to include in a collection, so that the total weight is less than a given limit and total value is as high as possible.* This combinatorial optimization problem would select supervoxels in a video based on their individual scores, hence

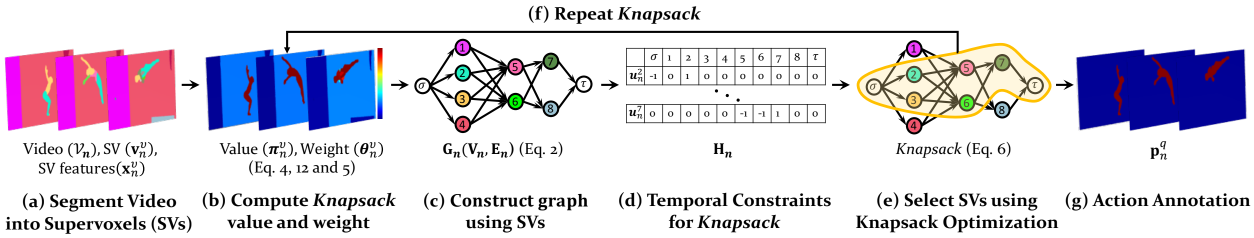

Figure 2. This figure shows the proposed *knapsack* approach in this paper. (a) Given an input video we extract supervoxel (SV) segmentation. (b) Each supervoxel is assigned a weight (spatio-temporal volume) and a value (score of belonging to the foreground action). (c) A graph $\mathbf{G}_n$ is constructed using supervoxels as nodes. (d) Temporal constraints are defined for the graph to ensure contiguous selection of supervoxels from start ($\boldsymbol{\sigma}$) to end ($\boldsymbol{\tau}$) of an action. (e) *Knapsack* optimization is applied to select a subset of supervoxels having maximum value, constrained by total weight (volume of the action) and temporal connectedness. (f) The *knapsack* process is repeated for more action annotations. (g) Annotations represented by action contours.

resulting in a degenerate solution, where selected supervoxels are not spatio-temporally connected throughout the video. Therefore, we propose a variant of *knapsack* problem with temporal constraints that enforces the annotated action to be well-connected and the weight limit ensures the detected volume is the size of an actor in the video. Since, the solution to the *knapsack* problem results in a single action annotation, we solve this problem iteratively to generate multiple annotations, while they satisfy the given constraints (see Fig. 2).

Let a video $\mathcal{V}_n$ be defined as a set of supervoxels $\mathcal{V}_n = \{\mathbf{v}_n^1, \mathbf{v}_n^2, \ldots, \mathbf{v}_n^M\}$, where $\mathbf{v}_n^{\upsilon}, \upsilon = 1, \ldots, M$ is the υth supervoxel in nth video and M is the total number of supervoxels in each video. The features associated with supervoxel $\mathbf{v}_n^{\upsilon}$ are given by $\mathbf{x}_n^{\upsilon} = \{{}_1\mathbf{x}_n^{\upsilon} \cdots {}_R\mathbf{x}_n^{\upsilon}\}$, where R is the total number of features. Next, we construct a *Directed Acyclic Graph (DAG)*, $\mathbf{G}_n(\mathbf{V}_n, \mathbf{E}_n)$ for each training video n, with supervoxels as nodes and edges connecting spatio-temporal neighbors. Graph $\mathbf{G}_n$ is a temporally forward flowing graph, that starts connecting supervoxels from the beginning of the video, to their temporal successors, until the end of the video. The adjacency matrix $\mathbf{Z}_n$ defining the graph $\mathbf{G}_n$ is as follows:

$$\mathbf{Z}_n(\upsilon, \upsilon') = \begin{cases} 1, & \text{if } \mathbf{v}_n^{\upsilon} \in \mathcal{N}_{\mathbf{G}_n}(\mathbf{v}_n^{\upsilon'}) \;\&\; f_{start}(\mathbf{v}_n^{\upsilon'}) > f_{start}(\mathbf{v}_n^{\upsilon}) \\ & \qquad \& \; f_{end}(\mathbf{v}_n^{\upsilon'}) > f_{end}(\mathbf{v}_n^{\upsilon}) \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $\mathcal{N}_{\mathbf{G}_n}(.)$ captures the spatio-temporal neighborhood of a supervoxel, f_{start} is the starting and f_{end} is the ending frame of a supervoxel.

Knapsack aims at selecting a contiguous and most valuable subset of nodes in this graph that form an action. Next, we define the value and weight of its items, as well as the temporal constraints.

Knapsack Value: Let the value of each supervoxel be defined by its score of belonging to the foreground action, $\boldsymbol{\pi}_n^{\upsilon}$. Each supervoxel in a video contains discriminative information towards an action, our aim is to assign every supervoxel an action distinctness score, which consists of: 1) *Humanness*, 2) *Saliency* [7] and 3) *Motion Boundary* [46].

Given a video $\mathcal{V}_n$, we use Faster-RCNN [31] to generate a set of human detection bounding boxes $\mathcal{B}_n = \{\mathcal{B}_n^1 \ldots \mathcal{B}_n^{F_n}\}$ along with their scores $\boldsymbol{\Gamma}_n = \{\boldsymbol{\Gamma}_n^1 \ldots \boldsymbol{\Gamma}_n^{F_n}\}$, where $\mathcal{B}_n^f$, $f = 1, \ldots, F_n$, is the set of bounding boxes in fth frame of a video $\mathcal{V}_n \in \mathcal{V}$ and F_n is the total number of frames. For each bounding box ${}_b\mathcal{B}_n^f \in \mathcal{B}_n^f$, the human detection score is given by ${}_b\boldsymbol{\Gamma}_n^f \in \boldsymbol{\Gamma}_n^f$. The *humanness* score for every supervoxel $\mathbf{v}_n^{\upsilon}$ is defined as:

$$\mathbf{S}_{hm}(\mathbf{v}_n^{\upsilon}, \mathcal{B}_n, \boldsymbol{\Gamma}_n) = \rho^{-1} \sum_{f=1}^{F_n} \arg\max_b \left(\frac{{}_f\mathbf{v}_n^{\upsilon} \cap {}_b\mathcal{B}_n^f}{|{}_f\mathbf{v}_n^{\upsilon}|} \right) . {}_b\boldsymbol{\Gamma}_n^f, \tag{3}$$

where ρ is the normalization factor, ${}_f\mathbf{v}_n^{\upsilon}$ is the segmented region in frame f and $|.|$ is its area. This function computes the weighted average overlap of a supervoxel region with its best overlapping bounding box in each frame.

We define the action distinctness as a combination of humanness, saliency and motion boundary as follows:

$$\boldsymbol{\Pi}(\mathbf{v}_n^{\upsilon}, \mathcal{B}_n, \boldsymbol{\Gamma}_n, \mathbf{x}_n^{\upsilon}) = \gamma_{hm}\mathbf{S}_{hm}(\mathbf{v}_n^{\upsilon}, \mathcal{B}_n, \boldsymbol{\Gamma}_n) + \gamma_{sal}\mathbf{S}_{sal}(\mathbf{v}_n^{\upsilon}, \mathbf{x}_n^{\upsilon}) + \gamma_{mb}\mathbf{S}_{mb}(\mathbf{v}_n^{\upsilon}, \mathbf{x}_n^{\upsilon}), \tag{4}$$

where $\mathbf{S}_{sal}(.)$ and $\mathbf{S}_{mb}(.)$ are the functions to compute supervoxel saliency and motion boundary, respectively. The associated weights in Eq. 4 are symbolized by γ. Finally, the supervoxel value is given by $\boldsymbol{\pi}_n^v = \mathbf{\Pi}(\mathbf{v}_n^v, \mathcal{B}_n, \mathbf{\Gamma}_n, \mathbf{x}_n^v)$.

Knapsack Weight: The weight of a supervoxel is defined by its spatio-temporal volume $\boldsymbol{\theta}_n^v$. We aim to select supervoxels that occupy a combined volume similar to that of an action. Hence, the total weight limit is defined as:

$$\mathbf{\Theta}_n = \mathcal{O}^{-1} \sum_{f=1}^{F_n} \sum_{b=1}^{\mathcal{O}} |\, {}_b\mathcal{B}_n^f|, \tag{5}$$

where $\mathcal{O}$ is the number of bounding boxes in each frame.

Temporal Constraints: We enforce temporal constraints to enable the algorithm in selecting supervoxels that are spatio-temporally connected. These constraints are defined on our *DAG*, to ensure that a supervoxel is selected only if at least one of its temporal predecessor is also selected. These set of constraints are defined by the rows of the matrix $\mathbf{H}_n = \mathbf{I} - \mathbf{Z}_n^T$, where $\mathbf{I}$ is the identity matrix and $\mathbf{Z}_n$ is from Eq. 2. Fig. 2(d) shows the rows of $\mathbf{H}_n$, whose sum should be less than or equal to zero for the selected supervoxels.

We associate with each supervoxel $\mathbf{v}_n^v \in \mathcal{V}_n$ a binary label variable $\mathbf{u}_n^v$, which is 1 if $\mathbf{v}_n^v$ belongs to the foreground action and 0 otherwise. In addition to the M supervoxel variables, we introduce two dummy variables: 1) *source* ($\mathbf{u}_n^\sigma$) and 2) *sink* ($\mathbf{u}_n^\tau$), that connect to the supervoxels in the first and last frame of a trimmed video, respectively. This ensures that the solution spans the entire length of the video. We solve the following *Binary Integer Linear Programming (BILP)* optimization to localize an action:

$$\begin{aligned} &\underset{\mathbf{u}_n}{\text{maximize}} \sum_{m=1}^{M+2} \boldsymbol{\pi}_n^m \mathbf{u}_n^m \text{ subject to } \sum_{m=1}^{M+2} \boldsymbol{\theta}_n^m \mathbf{u}_n^m \leq \mathbf{\Theta}, \\ &\mathbf{u}_n^m \in \{0,1\}, \quad \mathbf{u}_n^\sigma = 1, \quad \mathbf{u}_n^\tau = 1, \quad \mathbf{H}_n \mathbf{u}_n \leq 0. \end{aligned} \tag{6}$$

This function optimizes over supervoxels to select the set of supervoxels having maximum value, while satisfying temporal order and the weight limit. For untrimmed videos (e.g. in THUMOS13), we solve the above optimization by initiating and terminating at different temporal locations and sliding across the video.

Action Annotations: Since, each *knapsack* solution (in Eq. 6) gives an annotated action, we handle multiple actions in a video by recursively generating multiple annotations $\mathbf{p}_n^q = \bigcup_{v=1}^M \mathbf{v}_n^v, \quad \forall \mathbf{u}_n^v \neq 0$, where $q = 1, \ldots, Q_n$ and Q_n is the total number of action annotations in video $\mathcal{V}_n$, by excluding the selected supervoxels from $\mathcal{V}_n$ in each iteration.

4.1. Joint Annotation Selection

Action annotation using iterative *knapsack* approach can result in multiple action annotations per video, however due to complex background clutter, not all annotations may belong to the foreground action, due to false positives. Hence, we leverage multiple videos in a cluster C_k, to jointly select the annotations that belong to the common action class. The selected final action annotations per video, will be used to train a Support Vector Machine classifier and localize actions in testing videos.

We associate with each action annotation $\mathbf{p}_n^q$ a binary label variable $\mathbf{r}_n^q$, which is 1 if $\mathbf{p}_n^q$ contains the common action and 0 otherwise. We denote $\mathbf{r}_n$ to be a Q_n dimensional vector by stacking $\mathbf{r}_n^q$. Under the assumption that each video $\mathcal{V}_n$ has only one annotation that contains the common action, we solve the following *Binary Integer Quadratic Programming (BIQP)* optimization, which minimizes the distance between all action annotations across videos, under the constraint of selecting the most similar action annotation from each video:

$$\begin{aligned} &\underset{\mathbf{r}_n}{\text{minimize}} \quad \mathbf{r}_n^T \mathbf{U} \mathbf{r}_n - \mathbf{P} \mathbf{r}_n, \\ &\text{subject to} \quad \mathbf{r}_n \in \{0,1\}, \forall \mathcal{V}_n \in \mathcal{V} : \sum_{q=1}^{Q_n} \mathbf{r}_n^q = 1, \end{aligned} \tag{7}$$

where $\mathbf{U}$ is the χ^2 action distance matrix and $\mathbf{P}$ is the action annotation prior. For each action annotation $\mathbf{p}_n^q$, vector $\mathbf{P}$ contains the concatenated action prior score $\mathbf{\Phi}_n^q = \sum_{v=1}^M \boldsymbol{\pi}_n^v, \forall \mathbf{u}_n^v \neq 0$. Since the quadratic function in Eq. 7 is non-convex, we make it convex by taking the normalized laplacian [35] of $\mathbf{U}$, $\widetilde{\mathbf{U}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{U} \mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{D}$ is the diagonal matrix containing row sums of $\mathbf{U}$ and $\mathbf{I}$ is the identity matrix. Next, we relax the binary constraints of $\mathbf{r}_n$ to linear constraints, allowing it to take any value between 0 and 1, making it a convex optimization problem to be solved using standard techniques.

5. Unsupervised Action Localization

Given the automatically obtained action class labels and annotations for every training video, we learn a SVM action classifier to localize actions in testing videos. Next, we propose to use these annotations to discriminatively learn supervoxel unary and pairwise relations to compute action distinctness in *Knapsack*.

5.1. Training Action Classifiers

Knapsack approach to action annotation selects supervoxels by maximizing the sum of individual scores to annotate actions. These scores in Eq. 4 measure supervoxel distinctness based on their local features in an unsupervised manner. However, these scores are neither learnt discriminatively nor do they use information from neighboring relations in the graph $\mathbf{G}_n(\mathbf{V}_n, \mathbf{E}_n)$, to help select the best supervoxels. We propose to learn unary and pairwise scores from selected action annotations (see Sec. 4.1) in the training data.

SVM for Unary Learning: We learn a discriminative SVM classifier by using supervoxels within selected action annotations (in Eq. 7) as positive examples and the rest as negative examples.

Structural SVM for Pairwise Learning: Let $\mathbf{v}_n^{\upsilon'} \in \mathcal{N}_{\mathbf{G}_n}(\mathbf{v}_n^{\upsilon})$ belong to the neighborhood of $\mathbf{v}_n^{\upsilon}$. We gather such pairwise relations from training videos and their annotations (in Eq. 7) to propose a Structural Support Vector Machine (S-SVM) formulation with margin rescaling construction, which captures the relations between foreground-background as well as within foreground action using structured labels, as follows:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_{l=1}^{N*M} \xi_l, \\ \text{subject to} \quad & \langle \mathbf{w}, \boldsymbol{\Psi}_l([\mathbf{x}_l \mathbf{x}_{l'}], \mathbf{y}_l)\rangle - \langle \mathbf{w}, \boldsymbol{\Psi}_l([\mathbf{x}_l \mathbf{x}_{l'}], \mathbf{y})\rangle \\ & \qquad \geq \boldsymbol{\Delta}(\mathbf{y}_l, \mathbf{y}) - \xi_l, \\ & \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_l, \xi_l \geq 0, \forall l, \end{aligned} \tag{8}$$

where ξ represents the slack variables, $\mathbf{w}$ is the learned weight vector, $[.]$ is the concatenation of the feature vectors, $\mathcal{Y} = \{-1, 0, 1\}$ is the set of all labels and $\boldsymbol{\Psi}(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \text{sign}(\mathbf{y})$ is the joint feature function for a given input and output sample. The constraint with the loss function $\boldsymbol{\Delta}(\mathbf{y}_l, \mathbf{y})$ ensures that the score for the correct label $\mathbf{y}_l$ is higher than other labels. This can result in large number of constraints, therefore only a subset of constraints are used, known as the *most violated constraints*, by finding the label $\mathbf{y}$ which maximizes $\langle \mathbf{w}, \boldsymbol{\Psi}([\mathbf{x}_l \mathbf{x}_{l'}], \mathbf{y})\rangle + \boldsymbol{\Delta}(\mathbf{y}_l, \mathbf{y})$. The labels in Eq. 8 are defined as:

$$\mathcal{Y} = \begin{cases} -1, & \mathbf{v}^l \notin \boldsymbol{\kappa} \wedge \mathbf{v}^{l'} \notin \boldsymbol{\kappa} \\ 0, & \mathbf{v}^l \in \boldsymbol{\kappa} \wedge \mathbf{v}^{l'} \notin \boldsymbol{\kappa} \\ 1, & \text{otherwise}, \end{cases} \tag{9}$$

where $\boldsymbol{\kappa} = \bigcup_{n=1}^{N} \bigcup_{q=1}^{Q} \mathbf{p}_n^q$ and $\wedge$ is the logical AND operator. With ζ and ε as arbitrary constants showing relative loss, we define the loss function in Eq. 8 as:

$$\boldsymbol{\Delta}(\mathbf{y}_l, \mathbf{y}_{l'}) = \begin{cases} |\mathbf{y}_l - \mathbf{y}_{l'}|, & \mathbf{v}^l \notin \boldsymbol{\kappa} \wedge \mathbf{v}^{l'} \notin \boldsymbol{\kappa} \\ \zeta + \varepsilon, & \mathbf{v}^l \in \boldsymbol{\kappa} \wedge \mathbf{v}^{l'} \notin \boldsymbol{\kappa} \\ \varepsilon, & \text{otherwise}, \end{cases} \tag{10}$$

This loss function ensures that a pair of supervoxels get maximum score if they belong to the annotated action and minimum if either of them belongs to the background. A prediction function is learned $\psi_{\mathcal{P}} : \mathcal{X} \mapsto \mathcal{Y}$ that scores a pair of supervoxels in the testing video as:

$$\psi_{\mathcal{P}}([\mathbf{x}_t \mathbf{x}_{t'}]) = \arg\max_{y \in \mathcal{Y}} \langle \mathbf{w}, \boldsymbol{\Psi}_t([\mathbf{x}_t \mathbf{x}_{t'}], \mathbf{y}_t)\rangle. \tag{11}$$

5.2. Testing using Knapsack Localization

In a testing video $\mathcal{V}_s$, we compute supervoxels $\mathcal{V}_s = \{\mathbf{v}_s^1 \ldots \mathbf{v}_s^T\}$, where $t = 1, \ldots, T$, and extract their features $\mathbf{x}_s$ to build a *DAG*, $\mathbf{G}_s(\mathbf{V}_s, \mathbf{E}_s)$. Next, we apply *knapsack* approach (see Sec. 4 as used in training videos) along with SVM classifier, learned from automatically discovered video action class labels and annotations, to localize the action by solving the optimization in Eq. 6. Since, we are able to learn the unary and pairwise relations between supervoxels from action annotations in training videos, we use the following updated function to compute supervoxel action distinctness:

$$\begin{aligned} &\boldsymbol{\Pi}(\mathbf{v}_s^t, \mathcal{B}_s, \boldsymbol{\Gamma}_s, \mathbf{x}_s^t) = \gamma_{hm}\mathbf{S}_{hm}(\mathbf{v}_s^t, \mathcal{B}_s, \boldsymbol{\Gamma}_s) \\ &+ \gamma_{sal}\mathbf{S}_{sal}(\mathbf{v}_s^t, \mathbf{x}_s^t) + \gamma_{mb}\mathbf{S}_{mb}(\mathbf{v}_s^t, \mathbf{x}_s^t) + \gamma_{\mathcal{U}}\boldsymbol{\Upsilon}_{\mathcal{U}}(\mathbf{v}_s^t, \mathbf{x}_s^t) \\ &\qquad + \gamma_{\mathcal{P}}\boldsymbol{\Upsilon}_{\mathcal{P}}(\mathbf{v}_s^t, \mathbf{x}_s^t, \mathbf{G}_s), \end{aligned} \tag{12}$$

where $\Upsilon_{\mathcal{U}}(.)$ and $\Upsilon_{\mathcal{P}}(.)$ are the unary and pairwise functions, respectively. The weights in Eq. 12 are symbolized by γ. The pairwise function is an accumulation of neighboring relations $\boldsymbol{\Upsilon}_{\mathcal{P}}(\mathbf{v}_s^t, \mathbf{x}_s^t, \mathbf{G}_s) = \varrho^{-1} \sum_{t'=1}^{\mathcal{N}_{\mathbf{G}_n}(\mathbf{v}_s^t)} \psi_{\mathcal{P}}([\mathbf{x}_t \mathbf{x}_{t'}])$, where ϱ is a normalizing constant.

6. Experimental Results and Analysis

We evaluate our *Unsupervised Action Discovery and Localization* approach on five challenging datasets: 1) UCF Sports [32, 39] 2) JHMDB [11], 3) Sub-JHMDB [11] 4) THUMOS13 [13], and 5) UCF101 [40]. We provide the experimental setup, evaluation metrics, and an analysis of quantitative and qualitative results.

Experimental Setup: For the videos in training we extract C3D deep learning features [42] to cluster them into action classes with η=2. For action localization, we extract improved dense trajectory features (iDTFs) [46] for all videos. This is followed by supervoxel segmentation [25], which are encoded using Fisher [30] representation of iDTFs, with 256 Gaussians. *Knapsack* localization is classified using one-vs-all SVMs trained on action classes discovered by our approach. The parameters for *knapsack* value in Eqs. 4 and 12

Table 1. This table shows action discovery results using C3D on training videos of: 1) UCF Sports 2) Sub-JHMDB, 3) JHMDB, 4) THUMOS13, and 5) UCF101. We also show comparison of C3D [42] and iDTF [46] features on UCF Sports.

	UCF Sports iDTF	*UCF Sports* C3D	*Sub JHMDB*	*JHMDB*	*THUMOS 13*	*UCF101*
K-Means	*34.9*	*64.4*	*41.1*	*40.4*	*62.1*	*45.4*
K-Medoids	*26.4*	*59.6*	*36.6*	*34.3*	*67.3*	*33.0*
S&M [35]	*44.2*	*63.2*	*45.9*	*37.9*	*54.4*	*7.8*
DS [28]	*53.3*	*66.1*	*37.3*	*31.1*	*33.9*	*19.2*
SC [23]	*59.4*	*76.5*	*48.7*	*49.5*	*80.2*	*51.6*
DAKM [15]	*60.9*	*78.5*	*52.2*	*50.2*	*82.5*	*37.1*
Proposed	***69.9***	***90.1***	***57.4***	***53.7***	***88.3***	***61.2***

do not require tuning as we use normalized scores i.e. ($\gamma_{hm} = \gamma_{sal} = \gamma_{mb} = \gamma_{\mathcal{U}} = \gamma_{\mathcal{P}} = 1$). We used IBM CPLEX to solve BILP and BIQP optimizations.

Evaluation Metrics: Lan *et al*.'s [18] experimental setup is used to report localization results with Area Under Curve (AUC) of ROC (Receiver Operator Characteristic) at varying overlap threshold with the ground truth.

Unsupervised Action Discovery: The proposed approach discovers the action labels in training videos of five datasets. We compare the performance of our approach with: K-Means, K-Medoids, Shi and Malik (S&M) [35], Dominant Sets (DS) [28], Spectral Clustering (SC) [23] and the state-of-the-art DAKM [15] clustering methods. We follow DAKM's [15] experimental setup and evaluation, by setting the number of clusters to be the number of action classes in each dataset. The clustering results are reported in Table 1. Clustering on all datasets has been performed using C3D features, except for UCF Sports where we also report results using iDTF features for comparison. Table 1 shows that our approach gives superior performance on all five datasets. It is evident that unsupervised clustering of human actions is a challenging problem and known techniques such as *K-Means*, *K-Medoids* and NCuts [35] don't perform well. Significant improvement over *Dominant Sets* [28] and Spectral Clustering [23] highlights the strength of the proposed iterative approach, which we attribute to the ability of *dominant sets* to select a subset of coherent videos to train SVM and discriminatively learn to cluster actions. We observe highest performance on UCF Sports, which has the presence of distinct scenes and motion in the dataset, as compared to JHMDB and UCF101, that have complex human motion, independent of scene, and large intra-class variability.

Unsupervised Action Annotation: We independently evaluate the quality of annotations to localize actions, by assuming perfect action class labels to propose a weakly-supervised approach. We show the strength of

Table 2. This table shows comparison of localization performance with weakly-supervised approach [21] on UCF Sports.

Actions	Dive	Golf	Kick	Lift	Ride	
Ma et al. *[21]*	*44.3%*	*50.5%*	***48.3%***	*51.4%*	***30.6%***	
Proposed (Weakly)	***59.4%***	***59.9%***	*37.7%*	***59.5%***	*14.1%*	
Actions	Run	Skate	Swing-B	Swing-S	Walk	**Average**
Ma et al. *[21]*	*33.1%*	*38.5%*	***54.3%***	*20.6%*	*39.0%*	*41.0%*
Proposed (Weakly)	***50.0%***	***57.9%***	*50.0%*	***44.6%***	***43.4%***	***47.7%***

our *Knapsack* annotation approach by performing significantly better ($\sim 7\%$) than published state-of-the-art weakly-supervised method of Ma *et al*. [21] in Table 2.

Unsupervised Action Localization: We show localization performance using AUC curves for (a) UCF Sports (b) JHMDB, (c) Sub-JHMDB, and (d) THUMOS13 in Fig. 3. The difference in performance is attributed to the *supervised* vs. *unsupervised* nature of the methods. The results highlight that the proposed method performs competitive to the state-of-the-art supervised methods, that use video level class labels as well as ground truth bounding box annotations. In comparison we don't use any such information, and with our action discovery approach and *knapsack* for localization, we are able to perform better than some of the *supervised* methods [18, 41] on UCF Sports. *Supervised* baseline results have been reported by Wang *et al*. [47] on Sub-

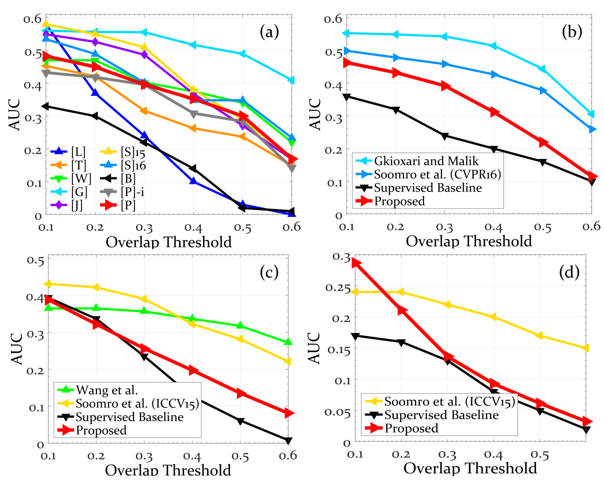

Figure 3. This figure shows AUC of the proposed *Unsupervised Action Localization* approach, along with existing *supervised* methods on (a) UCF Sports, (b) JHMDB, (c) Sub-JHMDB and (d) THUMOS13. The curves for the [P]roposed method is shown in red and *supervised* [B]aseline in black, while other *supervised* localization methods including [L]an *et al*. [18], [T]ian *et al*. [41], [W]ang *et al*. [47], [G]kioxari and Malik [6], [J]ain *et al*. [9], [S]oomro *et al*. [38, 36] are presented with different colors. For UCF Sports we also report our proposed ([P]-i) localization approach by learning a classifier on action discovery using iDTF [46] features.

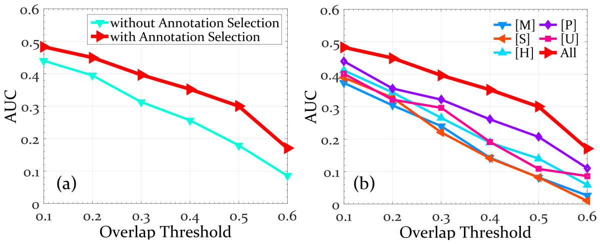

Figure 4. This figure shows the contribution of (a) Joint Annotation Selection (Eq. 7) and (b) individual components in computing action distinctness score for *Knapsack* value (Eq. 12) using AUC on UCF Sports. It includes [M]otion Boundary, [S]aliency, [H]umanness, [P]airwise S-SVM, [U]nary SVM and a combination of All i.e. M+S+H+U+P.

JHMDB and Soomro *et al*. [38, 36] on UCF Sports, JHMDB and THUMOS13. These baselines have been computed by exhaustively generating bounding boxes and connecting them spatio-temporally. Then, a classifier trained on ground truth annotations and iDTF features is applied for recognition. We outperform these baselines on all datasets in an *unsupervised* manner and at higher overlap thresholds. Our qualitative results are shown in Fig. 5, with action localization (yellow) and ground truth (green bounding box). In case of low-contrast and slow-motion the underlying supervoxel approach merges the actor with the background, therefore, when *knapsack* limits the localization to a specific actor volume, the proposed approach fails to localize (Fig. 5).

Feature Comparison: We show a comparison of the proposed *action discovery* approach using C3D and iDTF features in Table 1. The proposed approach performs significantly better using either features. C3D provides higher accuracy as they are semantically separable and provide better generalization over iDTF. Please note that although C3D features are extracted by supervised training on Sports1M dataset [42], we stress that these features are unsupervised relative to our experimental datasets, as no video action class label information nor bounding box annotations, from these datasets, have been used for feature training. Furthermore, we extend our comparison of features to *Unsupervised Action Localization* on UCF Sports (see Fig. 3 (a)). The results show similar localization performance of proposed approaches ([P] with C3D and [P]-i with iDTF), indicating the efficiency of *Knapsack* method for action detection.

Component's Contribution: The proposed approach has several steps that contribute to its performance. We quantify the relative contributions of each step in Fig. 4, which shows the AUC curves computed on UCF-Sports. In the absence of bounding box annotations, we use *knapsack* to annotate actions in training videos.

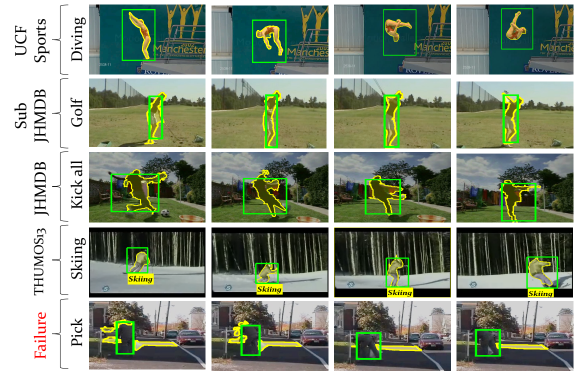

Figure 5. This figure shows qualitative results for the proposed approach on UCF Sports, Sub-JHMDB, JHMDB, and THUMOS13 datasets (top four rows). Last row shows failure case from JHMDB dataset. The action localization is shown by yellow contour and ground truth bounding box in green.

However, annotations may include false positives, resulting in a poorly trained classifier. Therefore, the *Annotation Selection* approach jointly selects action annotations in training videos that belong to the common action class within a cluster, to improve testing performance as shown in Fig. 4(a). Fig. 4(b) shows the contribution of each component in action distinctness score (Eq. 12), where pairwise learning using Structural SVM gives the best individual performance, capturing the supervoxel relations within a video to localize the action.

Computation Cost: *Knapsack* complexity is: $O(Mlog\frac{\Theta}{M})$. Total time for UCF Sports dataset: Action Discovery in Alg.1 ($\sim$2min), *Knapsack* in Eq.6 ($\sim$1min) and Joint Annotation Selection in Eq.7 ($\sim$1.1min), using an unoptimized MATLAB code running on an Intel Xeon E5645@2.4 Ghz/40GB RAM.

7. Conclusion

In this paper, we automatically discovered video action class labels and bounding box annotations to address the new problem of *Unsupervised Action Localization*. The presented approach discovers action classes, by using a discriminative clustering approach, and localizes actions, using novel *knapsack* optimization for supervoxel selection.

Acknowledgments: We would like to thank the anonymous reviewers for their useful feedback. The project was supported by Award No. 2015-R2-CX-K025, awarded by the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect those of the Department of Justice.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 2

[2] W. Chen and J. J. Corso. Action detection by implicit intentional motion clustering. In *ICCV*, 2015. 2

[3] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars. Online action detection. In *ECCV*, 2016. 2

[4] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV*. 2012. 2

[5] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2

[6] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015. 1, 2, 7

[7] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, 2006. 4

[8] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *ICCV*, 2009. 2

[9] M. Jain, J. Gemert, H. Jegou, P. Bouthemy, and C. Snoek. Action localization with tubelets from motion. In *CVPR*, 2014. 2, 7

[10] M. Jain, J. C. van Gemert, and C. G. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, 2015. 2

[11] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, 2013. 6

[12] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes. In *ECCV Workshop*, 2014. 2

[13] Y.-G. Jiang, J. Liu, A. Roshan Zamir, I. Laptev, M. Piccardi, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/ICCV13-Action-Workshop/, 2013. 6

[14] S. Jones and L. Shao. A multigraph representation for improved unsupervised/semi-supervised learning of human actions. In *CVPR*, 2014. 2

[15] S. Jones and L. Shao. Unsupervised spectral dual assignment clustering of human actions in context. In *CVPR*, 2014. 2, 7

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 2

[17] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ICCV*, 2007. 2

[18] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV*, 2011. 1, 7

[19] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli. Hierarchical clustering multi-task learning for joint human action grouping and recognition. 39(1), 2017. 2

[20] J. Lu, R. Xu, and J. J. Corso. Human action segmentation with hierarchical supervoxel consistency. In *CVPR*, 2015. 2

[21] S. Ma, J. Zhang, N. Ikizler-Cinbis, and S. Sclaroff. Action recognition and localization by hierarchical space-time segments. In *ICCV*, 2013. 2, 7

[22] S. Ma, J. Zhang, S. Sclaroff, N. Ikizler-Cinbis, and L. Sigal. Space-time tree ensemble for action recognition and localization. *IJCV*, 2017. 2

[23] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *NIPS*, 2002. 3, 7

[24] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3), 2008. 2

[25] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-temporal object detection proposals. In *ECCV*, 2014. 6

[26] D. Oneata, J. Verbeek, and C. Schmid. Efficient action localization with approximately normalized fisher vectors. In *CVPR*, 2014. 2

[27] M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *CVPR*, 2003. 3

[28] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *IEEE TPAMI*, 29(1), 2007. 3, 7

[29] X. Peng and C. Schmid. Multi-region two-stream r-cnn for action detection. In *ECCV*, 2016. 2

[30] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 6

[31] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 4

[32] M. Rodriguez, A. Javed, and M. Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 6

[33] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei. Spatial-temporal correlatons for unsupervised action classification. In *WMVC*, 2008. 2

[34] N. Shapovalova, M. Raptis, L. Sigal, and G. Mori. Action is in the eye of the beholder: Eye-gaze driven model for spatio-temporal action localization. In *NIPS*, 2013. 2

[35] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8), 2000. 5, 7

[36] K. Soomro, H. Idrees, and M. Shah. Action localization in videos through context walk. In *ICCV*, 2015. 2, 7, 8

[37] K. Soomro, H. Idrees, and M. Shah. Online localization and prediction of actions and interactions. *arXiv preprint arXiv:1612.01194*, 2016. 2

[38] K. Soomro, H. Idrees, and M. Shah. Predicting the where and what of actors and actions through online action localization. In *CVPR*, 2016. 7, 8

[39] K. Soomro and A. R. Zamir. Action recognition in realistic sports videos. In *Computer Vision in Sports*, pages 181–208. Springer, 2014. 6

[40] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2, 6

[41] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013. 2, 7

[42] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 3, 6, 7, 8

[43] D. Tran and J. Yuan. Max-margin structured output regression for spatio-temporal action localization. In *NIPS*, 2012. 2

[44] G. Vaca-Castano, S. Das, J. P. Sousa, N. D. Lobo, and M. Shah. Improved scene identification and object detection on egocentric vision of daily activities. *CVIU*, 156, 2017. 2

[45] J. C. van Gemert, M. Jain, E. Gati, and C. G. Snoek. Apt: Action localization psroposals from dense trajectories. In *BMVC*, 2015. 2

[46] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1, 4, 6, 7

[47] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *ECCV*. 2014. 2, 7

[48] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In *CVPR*, 2006. 2

[49] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, 2015. 2

[50] Y. Xie, H. Chang, Z. Li, L. Liang, X. Chen, and D. Zhao. A unified framework for locating and recognizing human actions. In *CVPR*, 2011. 2

[51] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE TPAMI*, 35(7), 2013. 2

[52] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. *arXiv preprint arXiv:1511.06984*, 2015. 2

[53] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016. 2

[54] G. Yu, N. A. Goussies, J. Yuan, and Z. Liu. Fast action detection via discriminative random forest voting and top-k subvolume search. *IEEE Transactions on Multimedia*, 13(3), 2011. 2

[55] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *CVPR*, 2015. 2

[56] J. Yuan, Z. Liu, and Y. Wu. Discriminative video pattern search for efficient action detection. *IEEE TPAMI*, 33(9), 2011. 2

[57] Z. Zhou, F. Shi, and W. Wu. Learning spatial and temporal extents of human actions for action detection. *IEEE Transactions on Multimedia*, 17(4), 2015. 2