ONLINE, SUPERVISED AND UNSUPERVISED ACTION LOCALIZATION IN VIDEOS

by

KHURRAM SOOMRO
M.S. Lahore University of Management Sciences, 2011
B.Sc (Hons) Lahore University of Management Sciences, 2007

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2017

Major Professor: Mubarak Shah

ABSTRACT

Action *recognition* classifies a given video among a set of action labels, whereas action *localization* determines the location of an action in addition to its class. The overall aim of this dissertation is action localization. Many of the existing action localization approaches exhaustively search (spatially and temporally) for an action in a video. However, as the search space increases with high resolution and longer duration videos, it becomes impractical to use such sliding window techniques. The first part of this dissertation presents an efficient approach for localizing actions by learning contextual relations between different video regions in training. In testing, we use the context information to estimate the probability of each supervoxel belonging to the foreground action and use Conditional Random Field (CRF) to localize actions. In the above method and typical approaches to this problem, localization is performed in an offline manner where all the video frames are processed together. This prevents timely localization and prediction of actions/interactions - an important consideration for many tasks including surveillance and human-machine interaction. Therefore, in the second part of this dissertation we propose an online approach to the challenging problem of localization and prediction of actions/interactions in videos. In this approach, we use human poses and superpixels in each frame to train discriminative appearance models and perform online prediction of actions/interactions with Structural SVM. Above two approaches rely on human supervision in the form of assigning action class labels to videos and annotating actor bounding boxes in each frame of training videos. Therefore, in the third part of this dissertation we address the problem of unsupervised action localization. Given unlabeled videos without annotations, this approach aims at: 1) Discovering action classes using a discriminative clustering approach, and 2) Localizing actions using a variant of Knapsack problem.

EXTENDED ABSTRACT

Action *recognition* involves classification of a given video in terms of a set of action labels, whereas action *localization* determines the location of an action in addition to its class. Many of the existing action localization approaches exhaustively search (spatially and temporally) for an action in a video. However, as the search space increases with high resolution and longer duration videos, it becomes impractical to use such sliding window techniques. The first part of this dissertation presents an efficient approach for localizing actions by learning contextual relations, in the form of relative locations between different video regions. We begin by over-segmenting the videos into supervoxels, which have the ability to preserve action boundaries and also reduce the complexity of the problem. Context relations are learned during training which capture displacements from all the supervoxels in a video to those belonging to foreground actions. Then, given a testing video, we select a supervoxel randomly and use the context information acquired during training to estimate the probability of each supervoxel belonging to the foreground action. The walk proceeds to a new supervoxel and the process is repeated for a few steps. A Conditional Random Field (CRF) is then used to find action proposals in the video, whose confidences are obtained using SVMs.

In the above method and typical approaches to this problem, localization or recognition is performed in an offline manner where all the frames in the video are processed together. This prevents timely localization and prediction of actions and interactions - an important consideration for many tasks including surveillance and human-machine interaction. Therefore, in the second part of this dissertation we propose a person-centric and online approach to the challenging problem of localization and prediction of actions and interactions in videos. In this approach, we estimate human poses in each frame and train discriminative appearance models using the superpixels inside the pose bounding boxes. Since the pose estimation per frame is inherently noisy, the conditional probability of pose hypotheses at current time-step (frame) is computed using pose estimations in the present frame and their consistency with poses in the previous frames. Next, both the superpixel and pose-based foreground likelihoods are used to infer the location of actors at each time through a CRF enforcing spatio-temporal smoothness in color, optical flow, motion boundaries and edges among superpixels. The issue of visual drift is handled by updating the appearance models, and refining poses using motion smoothness on joint locations, in an online manner. For online prediction of action (interaction) confidences, we propose an approach based on Structural SVM that operates on short video segments, and is trained with the objective that confidence of an action or interaction increases as time progresses.

Above two approaches rely on human *supervision* in the form of assigning action class labels to videos and annotating actor bounding boxes in each frame of training videos. Therefore, in the third part of this dissertation we address the problem of unsupervised action localization. Given unlabeled videos without annotations, this approach aims at: 1) Discovering action classes and 2) Localizing 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. Next, a discriminative clustering approach is applied by training a classifier for each cluster and videos are iteratively selected from the non-dominant set and complete video action classes are obtained. Annotations for training videos are obtained by over-segmenting videos into supervoxels and constructing a directed graph to apply a variant of knapsack problem. Knapsack selects supervoxels to generate action detections for each video. Within each cluster of videos, similar action detections are selected to train our action classifier. 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 validate the above proposed approaches on several challenging action datasets and show the action localization performance using standard metrics. Lastly, we also introduce UCF101 which

is one of the largest dataset of human actions. It consists of 101 action classes, over 13k clips and 27 hours of video data.The database consists of realistic user-uploaded videos containing camera motion and cluttered background.

To my parents,

and my brother,

for their lifelong love,

encoragement and sacrifices.

∼

To my lovely wife,

for her endless patience,

and support

∼

To my newborn son

for bringing joy

to our lives.

ACKNOWLEDGMENTS

All praises are due to Allah, the only worthy of worship, the most gracious and the most merciful. I would like to thank my advisor, Dr. Mubarak Shah for giving me the opportunity to learn and embark on this Ph.D. journey. His guidance and support over the years has helped me make progress in academic research. For this dissertation, I would like to thank my committee members Dr. Mark Heinrich, Dr. Ulas Bagci, Dr. Haiyan Hu and Dr. Hae-Bum Yun, for their precious services and valuable comments on my research work. I would also like to thank all the past and present members of the Center for Research in Computer Vision (CRCV), especially Dr. Haroon Idrees, Dr. Amir R. Zamir, Salman Khokhar, Dr. Waqas Sultani, Sarfaraz Hussein, Dr. Gonzalo Vaca, Dr. Nasim Souly, Dr. Kishore Reddy, Dr. Enrique Ortiz, Dr. Afshin Dehghan, Dr. Dong Zhang, Harish RaviPrakash, Shayan Modiri, Shervin Ardeshir, Mahdi Kalayeh, Amir Mazaheri, Muhammad Abdullah Jamal, Dr. Niels Lobo, Cherry Place, Brittany Kaval and Tonya LaPrarie, for their support and good memories. I would like to thank all my former teachers and professors, without whom I would not have been here today. My special thanks goes to my wife and my brother who have continuously supported me during my good and bad times, with their love, encouragement and optimism. Finally, I am indebted to my parents for giving me the upbringing, with their selfless dedication and sacrifices, that I have come this far.

TABLE OF CONTENTS

LIST OF FIGURES xiii

LIST OF TABLES xxii

CHAPTER 1: INTRODUCTION 1

1.1 Supervised Action Localization using Context Walk 4

1.2 Online Action and Interaction Localization 6

1.3 Unsupervised Action Localization 9

1.4 UCF101 Action Dataset 10

1.5 Dissertation Organization 10

CHAPTER 2: LITERATURE REVIEW 12

2.1 Supervised Action Analysis 13

2.1.1 Offline Action Localization 13

2.1.2 Online Action Prediction 16

2.1.2.1 Pose for Recognition 17

2.2 Unsupervised Action Analysis 19

CHAPTER 3: HUMAN ACTION DATASET 20

3.1 Dataset Details 21

3.2 Experimental Results 25

3.3 Related Datasets 28

3.4 Summary 28

CHAPTER 4: ACTION LOCALIZATION THROUGH CONTEXT WALK 30

4.1 Context Graphs for Training Videos 31

4.2 Context Walk in the Testing Video 32

4.2.1 Measuring Supervoxel Action Specificity 35

4.2.2 Inferring Action Locations using 3D-CRF 36

4.3 Experiments 37

4.3.1 Experiments on UCF-Sports 39

4.3.2 Experiments on THUMOS'13 40

4.3.3 Experiments on sub-JHMDB 40

4.3.4 Analysis and Discussion 41

4.4 Summary 46

CHAPTER 5: ONLINE LOCALIZATION AND PREDICTION OF ACTIONS AND IN-

TERACTIONS 47

5.1 Online Localization of Actions and Interactions 49

5.1.1 Superpixel-based Foreground Likelihood 50

5.1.2 Pose-based Foreground Likelihood 51

5.1.3 Actor Segmentation using Conditional Random Fields (CRF) 54

5.2 Online Prediction of Actions and Interactions 56

5.2.1 Binary SVMs with Dynamic Programming Inference (DP-SVM) 56

5.2.2 Structural SVM (S-SVM) 58

5.3 Experiments 59

5.3.1 Datasets 61

5.3.2 Results and Analysis 63

5.4 Summary 76

CHAPTER 6: UNSUPERVISED ACTION DISCOVERY AND LOCALIZATION 77

6.1 Action Discovery through Discriminative Clustering 78

6.2 Spatio-temporal Annotation of Training Videos using Knapsack 80

6.2.1 Joint Annotation Selection 85

6.3 Unsupervised Action Localization 86

6.3.1 Training Action Classifiers 86

6.3.2 Testing using Knapsack Localization 88

6.4 Experimental Results and Analysis 89

6.5 Summary 95

CHAPTER 7: CONCLUSION AND FUTURE WORK 96

LIST OF REFERENCES 99

LIST OF FIGURES

Figure 1.1: This figure illustrates the problem of *Online Action Localization* that we address in Chapter 5. The top row shows *kick ball* action being performed by a soccer player with frame number shown in top-left of each frame. The goal is to localize the actor (shown with yellow rectangles in top row) and predict the class label of the action (shown in red boxes in second row) as the video is streamed. As can be seen in the bottom row, the confidence of *kick ball* action increases and comes to the top as more of the action gets observed over time. This problem contrasts with *offline* action localization where action classification and detection is performed after the action or video clip has been observed in its entirety. 7

Figure 1.2: 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. The proposed method discovers actions by discriminative clustering using *dominant sets* and then applies a variant of *knapsack* problem to localize actions. 9

Figure 3.1: Sample frames for 6 action classes of UCF101. 21

Figure 3.2: 101 actions included in UCF101 shown with one sample frame. The color of frame borders specifies to which action type they belong: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, Sports. 22

Figure 3.3: Number of clips per action class. The distribution of clip durations is illustrated by the colors. 23

Figure 3.4: Total time of videos for each class is illustrated using the blue bars. The average length of the clips for each action is depicted in green. 24

Figure 3.5: Confusion table of baseline action recognition results using bag of words (BOW) approach on UCF101. The drawn lines separate different types of actions; 1-50: Sports, 51-60: Playing Musical Instrument, 61-80: Human-Object Interaction, 81-96: Body-Motion Only, 97-101: Human-Human Interaction. 27

Figure 4.1: This figure illustrates the idea of using context in the form of spatio-temporal displacements between supervoxels. (a) Given N_c videos for an action c which have been over-segmented into supervoxels, we construct a context graph for each video as shown in (b). Each graph has edges emanating from all the supervoxels to those that belong to foreground action (circumscribed with dashed green contours). The color of each node in (b) is the same as that of the corresponding supervoxel in (a). Finally, a composite graph (Ξ) from all the context graphs is constructed, implemented efficiently using a kd-tree. (c) We also quantify 'supervoxel action specificity' which returns the likelihood of a particular supervoxel belonging to an action and use it in conjunction with context to localize actions. 31

Figure 4.2: This figure depicts the testing procedure of the proposed approach. (a) Given a testing video, we perform supervoxel (SV) segmentation. (b) A graph $\mathbf{G}$ is constructed using the supervoxels as nodes. (c) We find the nearest neighbors of the selected supervoxel ($\mathbf{v}^{\tau}$; initially selected randomly) in the composite graph Ξ which returns the displacement vectors learned during training. The displacement vectors are projected in the testing video as shown with yellow arrows. (d) We update the foreground/action confidences of all supervoxels using all the NNs and their displacement vectors. (e) The supervoxel with the highest confidence is selected as $\mathbf{v}^{\tau+1}$. (f) The walk is repeated for T steps. (g) Finally, a CRF gives action proposals whose action confidences are computed using SVM. 34

Figure 4.3: This figure shows the average of maximum supervoxel overlap in every training video of different actions as a function of segmentation level. Using the correct level from the hierarchy reduces the number of potential supervoxels we have to handle while testing. This speeds up the method without sacrificing performance. 38

Figure 4.4: The ROC and AUC curves on UCF Sports Dataset [71] are shown in (a) and (b), respectively. The results are shown for Jain *et al.* [30] (orchid), Tian *et al.* [86] (blue), Lan *et al.* [46] (amber), Wang *et al.* [95] (green) and Proposed Method (red). (c) shows the AUC for THUMOS'13 dataset [33], for which we are the first to report results. 39

Figure 4.5: The ROC and AUC curves for sub-JHMDB dataset [32, 95] are shown in (a) and (b), respectively. Green and black curves are from the method by Wang *et al.* [95] and their iDTF + Fisher Vector baseline. Red curve shows the performance of the proposed method which is better than [95]. 41

Figure 4.6: This figure shows the contributions of the fours aspects of the proposed approach towards overall performance, in terms of ROC (left) and AUC of Precision-Recall curve as a function of overlap threshold (right). 43

Figure 4.7: This figure shows qualitative results of the proposed approach (yellow contours) against ground truth (green boxes) on selected frames of testing videos. The first four rows are from UCF-Sports [71], next four are from sub-JHMDB [95], and then the next four rows are from THUMOS'13 [33] datasets. Last two rows show two failure cases from sub-JHMDB dataset. 44

Figure 5.1: This figure shows the framework of the approach proposed in this chapter. (a) Given an input video, (b) we over-segment each frame into superpixels and (c) detect poses using an off-the-shelf method [99]. (d) An appearance model is learned using all the superpixels inside a pose bounding box as positive, and those outside as negative samples. (e) In a new frame, the appearance model is applied on each superpixel of the frame to obtain a foreground likelihood. (f) To handle the issue of visual drift, poses are refined using spatio-temporal smoothness constraints on motion and appearance. (g) Finally, a CRF is used to obtain local action proposals at each frame, (h) on which actions (interactions) are predicted through Structural SVM. 48

Figure 5.2: This figure shows a visualization of the joint smoothness costs used in pose-based foreground likelihood for (a) appearance smoothness of joints (J_{app}), (b) location smoothness of joints (J_{loc}) and (c) scale smoothness of joints (J_{sc}). 52

Figure 5.3: This figure shows online action (interaction) localization performance as a function of observed action percentage on (a) MSR-II and (b) UT-Interaction datasets. In contrast to Fig. 5.4(d,f), there are two important differences. First, mean average precision (mAP) is reported instead of multi-label prediction (classification). Furthermore, the evaluation is over untrimmed videos, and includes background clips. 63

Figure 5.4: This figure shows action prediction accuracy as a function of observed percentage of action or interaction for (a) UCF Sports, (b) JHMDB, (c) sub-JHMDB, (d) MSR-II, (e) TV Human Interaction and (f) UT Interaction datasets. Prediction performance by the baseline Binary SVM with Dynamic Programming approach is shown in blue, and that of Structural SVM with the red curve. We compare the performance of our action prediction with MMED [25] (yellow curve) for UCF Sports and Sub-JHMDB datasets. 64

Figure 5.5: This figure shows per-action prediction accuracy as a function of observed action (interaction) percentage for (a) sub-JHMDB and (b) TV Human Interaction datasets. The mean accuracy for all actions (interactions) is shown with bold red curve. 66

Figure 5.6: This figure shows per-action prediction accuracy as a function of observed action percentage for UCF Sports dataset for (a) Structural SVM approach (Sec. 5.2.2) and (b) and its difference with SVM and Dynamic Programming (Sec. 5.2.1). On average, S-SVM outperforms DP-SVM. 67

Figure 5.7: This figure shows online action (interaction) localization performance as a function of observed action percentage on (a) JHMDB, (b) sub-JHMDB, (c) MSR-II, (d) UCF-Sports, and as a function of observed interaction percentage for (e) TV Human Interaction and (f) UT Interaction datasets. Different curves show evaluations at different overlap thresholds: 10% (red), 30% (green) and 60% (pink). 68

Figure 5.8: This figure shows action localization results of the baseline Binary SVM with Dynamic Programming (DP-SVM) and Structural SVM (S-SVM) approaches, along with existing *offline* methods on four action datasets (JHMDB, UCF Sports, sub-JHMDB and MSR-II). (a) shows AUC curves for JHMDB, while (b) and (c) show AUC and ROC @ 20%, respectively, for UCF Sports dataset. AUC and ROC @ 20% overlap are shown in (d) and (e) for sub-JHMDB dataset, finally AUC for MSR-II dataset is shown in (f). The curve for S-SVM method is shown in red and DP-SVM is shown in blue, while other *offline* localization methods including Lan *et al.* [46], Tian *et al.* [86], Wang *et al.* [95], van Gemert *et al.* [92], Jain *et al.* [30] [31], Gkioxari and Malik [20], Chen and Corso [11], Weinzaepfel *et al.* [102] and Soomro *et al.* [81] are shown with different colors. Despite being online, the proposed approach performs competitively overall compared to existing offline methods. 70

Figure 5.9: This figure shows action localization results on MSR-II dataset. The precision/recall curves are drawn for three actions: (a) boxing, (b) Hand clapping and (c) hand waving. We perform competitively to many existing *offline* methods. Red curve shows the proposed S-SVM approach, while blue curve shows the results of the baseline DP-SVM method. 71

Figure 5.10This figure shows interaction localization results on two interaction datasets. ROC @ 20% overlap and AUC curves for TV Human Interaction dataset are shown in (a) and (b), and for UT Interaction dataset in (c) and (d). In this figure, S-SVM shown in red and DP-SVM (baseline) in blue. 72

Figure 5.11This figure shows qualitative results for pose refinement. Results show a comparison of raw poses (top row) and refined poses (bottom row) for (a) Kicking and (b) Walking. 72

Figure 5.12This figure shows qualitative results of the proposed approach for JHMDB, sub-JHMDB and UCF Sports datasets, where each action segment is shown with yellow contour and ground truth with green bounding box. 74

Figure 5.13This figure shows qualitative results of the proposed approach for MSR-II, TV Human Interaction and UT Interaction datasets, where each action segment is shown with yellow contour and ground truth with green bounding box. 75

Figure 6.1: 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 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. 81

Figure 6.2: 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.* [46], [T]ian *et al.* [86], [W]ang *et al.* [95], [G]kioxari and Malik [20], [J]ain *et al.* [30], [S]oomro *et al.* [81, 82] 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 [94] features. 92

Figure 6.3: This figure shows the contribution of (a) Joint Annotation Selection (Eq. 6.7) and (b) individual components in computing action distinctness score for *Knapsack* value (Eq. 6.12) 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. 93

Figure 6.4: 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. 94

LIST OF TABLES

Table 3.1: Summary of Characteristics of UCF101 25

Table 3.2: Summary of Major Action Recognition Datasets 28

Table 4.1: Quantitative comparison of proposed approach with existing methods at 20% overlap. 42

Table 4.2: Number of classifier evaluations as a function of supervoxels / subvolumes in a video. 46

Table 5.1: This table shows the the percentage of video observation required to achieve a prediction accuracy of 30%. Results in the first two rows are from JHMDB, then from sub-JHMDB and the last row is from UCF Sports dataset. Actions with missing values indicate that they did not reach a prediction accuracy of 30% until video completion. 69

Table 5.2: This table shows the average precision for MSR-II dataset on three different actions: (a) Boxing, (b) Handclapping and (c) Handwaving. 73

Table 5.3: This table shows the video mean average precision (mAP) for UCF Sports, JHMDB, Sub-JHMDB, TV Human Interaction and UT Interaction datasets. . 73

Table 6.1: 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 [87] and iDTF [94] features on UCF Sports. 90

Table 6.2: This table shows comparison of localization performance with weakly-supervised approach [52] on UCF Sports. 91

CHAPTER 1: INTRODUCTION

The cognitive abilities of humans enables them to perceive, comprehend, analyze, and interact with objects in their surroundings. Among these skills, visual processing plays an important role in understanding the dynamics of the real world. Thus, making us capable of recognizing objects, faces and activities, as well as anticipating events in advance. The goal of computer vision and machine learning is to help computers in learning to imitate human perception. Every day people capture a large collection of videos and images using their cellphones, and share them on social media e.g. YouTube, Facebook, Instagram or Flickr. Millions of surveillance cameras around the world record billions of hours of video footage. With this large influx of Big Data, it has become impractical for humans to view and distill useful information from the collected data. Therefore, it is imperative that we develop algorithms for automatic analysis and understanding of videos.

The most challenging problems associated with automated analysis of videos are related to actions, with a variety of computer vision approaches [6, 101]. One of the problems is *action recognition* which entails classification of a given video in terms of a set of action labels. With the introduction of uncontrolled datasets, consisting of videos captured in realistic non-experimental settings and longer durations such as those from YouTube [33, 71], *action detection (or localization)* has emerged as a new problem where the goal is to determine the location of an action in addition to its class. Action detection, which may refer to temporal detection [33] or spatio-temporal action localization [12, 17, 30, 60], is especially difficult when background is cluttered, videos are untrimmed or contain multiple actors or actions.

Recognizing and localizing actions has been fundamental to video understanding in computer vision. It is a challenging problem, which has a wide variety of applications from monitoring and security in surveillance videos, to video search, action retrieval, multimedia event recounting [2] and human-computer interaction (HCI).

Many existing approaches [86, 95] learn an action detector on trimmed training videos and then exhaustively search for each action through the testing videos. However, with realistic videos having longer durations and higher resolutions, it becomes impractical to use sliding window approach to look for actions or interesting events [30, 46, 114]. Analyzing the videos of datasets used for evaluation of action localization such as UCF-Sports [71], JHMDB [32], and THUMOS [33] reveals that, on average, the volume occupied by an action (in pixels) is considerably small compared to the spatio-temporal volume of the entire video (around 17%, using ground truth). Therefore, it is important that action localization is performed through efficient techniques which can classify and localize actions without evaluating at all possible combinations of spatio-temporal volumes.

Existing *offline* action localization methods [30, 46, 81, 86, 95, 111] classify and localize actions after completely observing an entire video sequence. The goal is to localize an action by finding the volume that encompasses an entire action. Some approaches are based on sliding-windows [60, 86], while others segment the video into supervoxels which are merged into action proposals [30, 59, 81]. The action proposals from either methods are then labeled using a classifier. Essentially, an action segment is classified ***after*** the entire action volume has been localized. Similarly, the videos are processed for classification [26, 41], retrieval [61, 62] or localization [74] in an offline manner for the case of interactions. Since offline methods have entire video and action segments at their disposal, they can take advantage of observing entire motion of action instances, and for practical purposes do not provide action detection in a timely manner. Similarly, there have been recent efforts to predict activities by early recognition [43, 45, 47, 73]. However, these methods only attempt to predict the label of the action without any localization. Thus, the important question about *where* an action is being performed remains unanswered, which we tackle in this dissertation.

Current action localization approaches [20, 46, 94] 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 bounding box annotations and *recognize* using action class labels from training data. Since *supervised* methods have the annotated ground truth at their disposal, they can take the 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 consuming. Third, actions vary spatio-temporally (i.e., in height, width, spatial location, and temporal length) resulting in various tubelet deformations. Fourth, people may have different understandings of the temporal extent of an action, which results in 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 [4, 18, 34, 38, 84]. 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 dissertation, we aim to address the problem of action localization in videos with its recognized labels. We explore this problem in an ***offline vs. online*** and in a ***supervised vs. unsupervised*** setting. We summarize the following important contributions.

- We propose an efficient approach for action localization by learning contextual relations in the form of relative locations between different video regions (i.e. supervoxels). In a testing video, we select a supervoxel and use this learned information to perform *Context Walk*. This generates a conditional distribution of an action over all supervoxels and is used to localize an action. As a result, we significantly reduce the number of classifier evaluations, in sharp contrast to the alternate sliding window approaches.

- We introduce a new problem of *Online Action and Interaction Localization* and propose a novel person-centric approach that uses human pose estimation and superpixels in each frame to compute foreground likelihoods. Then, the superpixel and pose-based foreground likelihoods are used to infer the location of the actors at each time instant (frame) using a Conditional Random Field. For online prediction, we propose a Structural SVM based approach trained with the objective, that confidence of an action (interaction) should increase with time.
- We address a new problem of *Unsupervised Action Localization*, which aims at: 1) Discovering action classes, and 2) Localizing actions, given an unlabeled set of videos without bounding box annotations. We propose a novel approach for action discovery, that uses Dominant Sets and a discriminative clustering algorithm to iteratively select videos and obtain complete video action clusters. For localizing actions we propose a Knapsack formulation that enforces spatio-temporal constraints on supervoxels, to detect actions in a video.
- We introduce *UCF101*, a new action recognition and localization dataset. It contains 101 action classes, with over 13K video clips and 27 hours of video footage. We also generate an action recognition baseline on this new dataset using standard bag of words approach.

1.1 Supervised Action Localization using Context Walk

The use of context has been extensively studied for object detection in images through modeling the relationships between the objects and their surroundings [7, 14, 24], which significantly reduce search space of object hypotheses. However, it is non-trivial to extend such approaches to actions in videos, since the temporal dimension is very different from the spatial dimensions. An image or a video is confined spatially, but the temporal dimension can be arbitrarily long. The differences in spatial and temporal dimensions also affects the optimal representation of actions in videos [86]. Cuboid, which is the 3D extension of a bounding box in images, is not appropriate for action localization due to the following two reasons: (1) Actions have a variable aspect ratio in space and time as they capture articulation and pose changes of actors. Furthermore, instances of repetitive actions (such as running) can have different lengths depending on the number of cycles captured in the video. (2) The nature of an action or simply the camera motion can cause an actor to move spatially in a video as time progresses. In such a case, a cuboid would include large parts of the background. Accordingly, the ground truth in action localization datasets consists of a sequence of bounding boxes which change in size and move spatially with respect to time. Each such sequence can be visualized as a rectangular tube with varying height, width and spatial location.

On the same grounds, the results of action localization will be more useful if they contain minimal background, which cannot be achieved with cuboid or sliding window approaches [86, 89, 97, 114]. However, such a powerful representation of actions come with a cost. Generating tight tubes around the actors makes the task of action localization even more challenging as the action hypotheses not only depend on space and time, but also on tube deformations. An exhaustive search over all possible combinations is wasteful and impractical. In Chapter 4, we formulate the problem of action localization in such a way that the issues associated with cuboid and sliding window approaches are circumvented and use context to significantly reduce the search space of hypotheses resulting in fewer number of evaluations during testing.

For the proposed approach, we over-segment the videos into supervoxels and use context as a spatial relation between supervoxels relative to foreground actions. The relations are modeled using 3D displacement vectors which capture the intra-action (foreground-foreground) and action-to-scene (background-foreground) dependencies. These contextual relations are represented by a graph for each video, where supervoxels form the nodes and directed edges capture the spatio-temporal relations between them. During testing, we perform a context walk where each step is guided by the context relations learned during training, resulting in a probability distribution of an action over all the supervoxels.

There are a few approaches that reduce the search space to efficiently localize actions. To the best of our knowledge, we are the first to explicitly model foreground-foreground and background-foreground spatial relationships for action localization. The proposed approach requires only a few nearest neighbor searches in a testing video followed by a single application of CRF that gives action proposals. The action confidences of proposals are then obtained through SVM. This is in contrast to most of the existing methods [86, 95], which require classifier evaluations several order of magnitudes higher than the proposed approach.

1.2 Online Action and Interaction Localization

Predicting *what* and *where* an action or interaction occurs is an important and challenging computer vision problem for automatic video analysis [16, 46, 95, 111]. It involves the use of limited motion information in partially observed videos for frame-by-frame localization and label prediction, and has varied applications in many areas. For human-computer or human-robot interaction, it allows the computer to automatically localize and recognize actions and gestures as they occur, or predict the intention of actors, thereafter creating appropriate responses for them. It is especially relevant to the monitoring of elderly, where detection of certain actions, e.g. *falling*, must trigger an immediate automated response and alert the care giver or a staff member. Moreover, this allows their interactions with other people to be monitored and quantified for overall well-being.

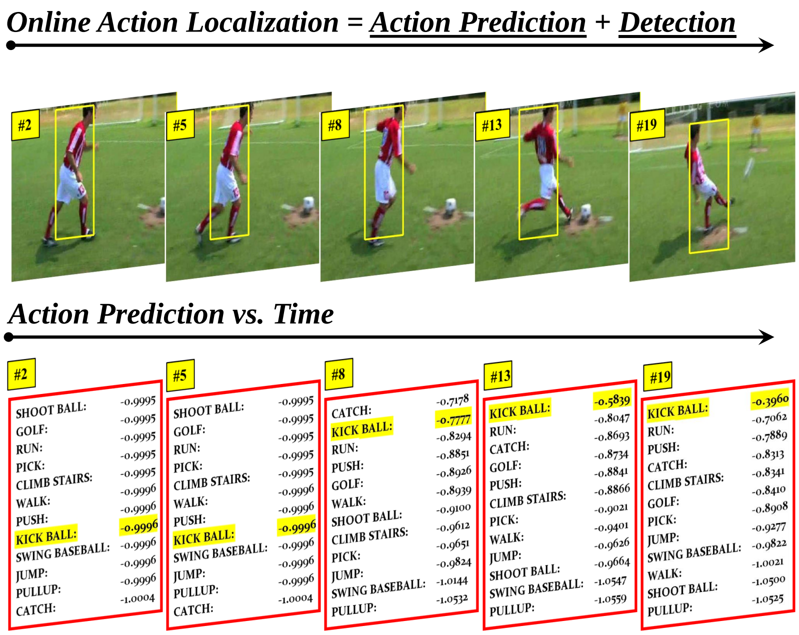


Figure 1.1: This figure illustrates the problem of *Online Action Localization* that we address in Chapter 5. The top row shows *kick ball* action being performed by a soccer player with frame number shown in top-left of each frame. The goal is to localize the actor (shown with yellow rectangles in top row) and predict the class label of the action (shown in red boxes in second row) as the video is streamed. As can be seen in the bottom row, the confidence of *kick ball* action increases and comes to the top as more of the action gets observed over time. This problem contrasts with *offline* action localization where action classification and detection is performed after the action or video clip has been observed in its entirety.

In visual surveillance, online localization and prediction can be used for detecting abnormal actions such as assault or interactions of criminal nature, e.g., drug exchange and alert the human monitors in a timely manner. In automated robot navigation or autonomous driving, the timely detection of human actions in the environment will lead to requisite alteration in path or speed, e.g., a child jumping in front of the car. In Chapter 5, we address the very problem of *Online Action and Interaction Localization*, which aims at localizing actions (interactions) and predicting their class labels in a streaming video (see Fig. 1.1).

In this work, for online action (interaction) localization and prediction, we propose to use the high level structural information using pose in conjunction with a superpixel based discriminative actor foreground model that distinguishes the foreground actor from the background. The superpixel-based model incorporates visual appearance using color and motion features, whereas the pose-based model captures the structural cues through joint locations. Using both the foreground and pose models we generate a confidence map, used to locate the action segments by inferring on a Conditional Random Field in an online manner. Since the appearance of an actor changes due to articulation and camera motion, we retrain foreground model as well as impose spatio-temporal smoothness constraints on poses to maintain representation that is both robust and adaptive. As soon as the human actors are localized at the current frame, we proceed to recognize and predict the label of the action (interaction). There can be multiple approaches to perform online prediction, since the windows over which the visual features are accumulated can be defined in various ways. In [82], we use a hybrid of binary SVM and dynamic programming on short intervals to predict the class labels in an online manner. However, this requires multiple classifiers to be trained for each sub-action or segment of an action. We present an alternate approach that uses a Structural SVM, trained with the objective that the score of the action (interaction) should increase as time passes in clips containing positive training instances. Finally, we perform rigorous experiments on four action and two interaction datasets, and introduce measures for consistent evaluation across both actions and interactions.

The contributions of Chapter 5 can be summarized as follows: 1) We address the problem of *Online Action and Interaction Localization* in streaming videos, 2) by using high-level pose estimation to learn mid-level superpixel-based foreground models at each time instant. 3) We employ spatio-temporal smoothness constraints on joint locations in human poses to obtain stable and robust action segments in an online manner. 4) The label and confidences for action (interactions) segments are *predicted* using Structural SVM trained on partial action (interaction) clips, which enforces the constraint that the confidence of positive samples increases monotonically over time.

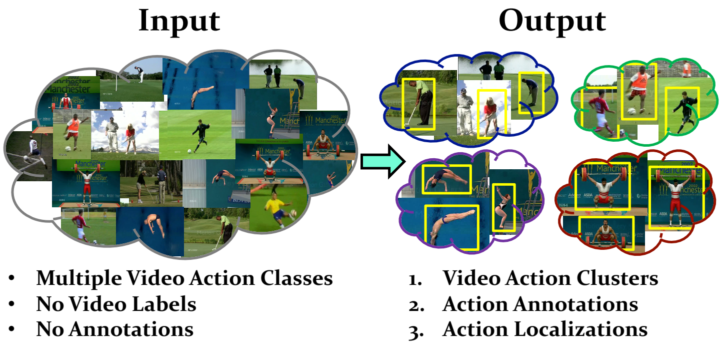


Figure 1.2: We tackle the problem of *unsupervised action localization* without any action class labels or bounding box annotations, where a given collection of unlabeled videos contain multiple action classes. The proposed method discovers actions by discriminative clustering using *dominant sets* and then applies a variant of *knapsack* problem to localize actions.

Finally, 5) we introduce an evaluation measure to quantify performance of action (interaction) prediction and online localization and perform experiments on six action and interaction datasets with a consistent evaluation framework.

1.3 Unsupervised Action Localization

The problem of *Unsupervised Action Localization* aims at localizing an action without the use of ground truth in training videos (see Fig. 1.2). The training data for action localization usually provides: 1) action classes and 2) actor bounding boxes. In chapter six of this thesis, we automatically discover action classes by discriminatively clustering a group of unlabeled 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 action localization. In this approach, we segment the video into supervoxels and in a combinatorial optimization framework we select the supervoxels that belong to the actor performing the action.

In summary, Chapter 6 makes the following contributions: 1) We address the problem of *Unsupervised Action Localization*, 2) by proposing a discriminative clustering approach using *dominant sets* to discover action classes. 3) We propose *knapsack* approach with graph-based temporal constraints on supervoxels to obtain action localization in an *unsupervised* manner. 4) The localizations within each cluster of videos are jointly selected to train action classifiers and finally, 5) 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.

1.4 UCF101 Action Dataset

Action recognition and localization algorithms require benchmarking their performance on public datasets. Majority of existing datasets have two disadvantages: 1) Low number of action classes and 2) Videos recorded in an unrealistically controlled environment. Therefore, we propose a human action dataset with 101 action classes and 13320 videos, collected from YouTube in an unconstrained environment. These are challenging videos with camera motion, various lighting conditions, partial occlusion and varying quality of video frames. This dataset consists of manually annotated video action class labels, as well as frame level bounding box annotations for 24 human action classes.

1.5 Dissertation Organization

The rest of the dissertation is structured as follows: In Chapter 2, we review existing literature on action recognition and localization. In Chapter 3, we present a new human action dataset with 101 action classes. Chapter 4 proposes an efficient action localization approach by learning contextual relations in videos. Chapter 5 introduces a new problem of *Online Action and Interaction Localization* using a novel person-centric approach. Chapter 6 addresses a new problem of *Unsupervised Action Localization* containing two sub-problems of action discovery and localization.

CHAPTER 2: LITERATURE REVIEW

Action recognition in realistic videos has been an active area of research with several recent surveys [6, 101] published on the subject. With significant progress in action recognition over the past few years, researchers have now started focussing on the more difficult problem of action localization [17, 27, 31, 60, 95, 106, 111]. This involves simultaneous detection and recognition of actions. The detection can be spatial, by placing a bounding box over the actor in each frame or spatio-temporal, which also requires determining the temporal extent of the action (i.e. starting and ending frame). Until recently, majority of the published research in action localization has been focused towards *supervised* learning. This requires lots of training videos, which have been manually labeled with video action class and each frame has been annotated with actor bounding boxes. When an action detector is learned using this training video set, it is applied on a testing video using an *offline* algorithm. This assumes that the entire frames of a testing video are provided, and an exhaustive sliding window/volume technique is applied to localize an action.

The focus of this dissertation is to solve the problem of action localization. Firstly, we propose a *supervised offline* approach that avoids using costly exhaustive sliding window approach to localize the action in an *efficient* manner (see Chapter 4). Next, we address the limitation of *supervised offline* approaches by localizing actions in a *supervised online* and timely manner (see Chapter 5). Finally, we also show that we don't necessarily require ground truth for video action class labels or bounding box annotations, by localizing actions in an *unsupervised* manner (see Chapter 6).

In this chapter, we cover existing literature for action localization, by reviewing *supervised* and *unsupervised* action approaches. We begin with *supervised offline* approaches that require ground truth for training and perform *offline* testing. These include approaches that use search based methods for action localization, and *context* for recognition. After that, we discuss *online* approaches for action prediction. Finally, we cover *unsupervised* action techniques that mainly involve action clustering.

2.1 Supervised Action Analysis

Supervised action localization has been extensively studied in recent years [11, 17, 27, 31, 39, 51, 65, 79, 95, 102, 106, 110, 111, 112]. This section covers the works that use video action class labels and bounding box annotations to train action classifiers and detectors in their proposed methods. However, these methods can further be distinguished into 1) *offline testing* and 2) *online testing*. In the following sub-sections we first cover *offline* approaches and then *online* methods.

2.1.1 Offline Action Localization

Offline localization has received significant attention in the past few years, both for actions [17, 27, 31, 106] as well as interactions [40, 72]. For actions, the first category of approaches uses either rectangular tubes or cuboid-based representations to exhaustively search for an action using a sliding volume approach. Lan *et al.* [46] treated the human position as a latent variable, which is inferred simultaneously while localizing an action. Yuan *et al.* [114] used branch-and-bound with dynamic programming, while Zhou *et al.* [117] used a split-and-merge algorithm to obtain action segments that are then classified with LatentSVM [19]. Wang *et al.* [95] tackle the problem of action detection with poselet estimation. Oneata *et al.* [60] presented an approximation to Fisher Vectors for tractable action localization. Tran and Yuan [88] used Structural SVM to localize actions with inference performed using Max-Path search method. Ma *et al.* [52] automatically discovered spatio-temporal root and part filters, whereas Tian *et al.* [86] developed Spatio-temporal Deformable Parts Model [19] to detect actions in videos and use a sliding window approach to handle deformities in parts, both in space and time. Recently, Yu and Yuan [112] proposed a method for generating action proposals obtained by detecting tubes with high actionness scores after non-maximal suppression.

The second category uses either superpixels or supervoxels as the base representations [30, 59]. Such representation helps reduce the search space from pixel-level to superpixel or supervoxel level. Jain *et al.* [30] recently proposed a method that extends selective search approach [90] to videos, where they merge supervoxels using appearance and motion costs and produce multiple layers of segmentation for each video. Soomro *et al.* [81] uses context walk with Conditional Random Field (CRF) to segment actions. These supervoxel based methods 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. Gkioxari and Malik [20] use selective search [90] to generate candidate proposals for video frames, whose spatial and temporal (motion) Convolutional Neural Network (CNN) features are evaluated using SVMs. The per-frame action detections are then linked temporally for localization. There have been few similar recent methods for quantifying actionness [12, 112], which yield fewer regions of interest in videos. For interaction recognition in videos, Kong *et al.* [41, 42] learn high-level descriptions called interactive phrases to express binary semantic motion relationships between interacting people. A hierarchical model is used to encode interactive phrases based on latent SVM framework where interactive phrases are treated as latent variables. Wu *et al.* [103] also decompose interaction video segments into spatial cells and learn relationship between them. In Chapter 6, our *knapsack* approach is different from the above supervoxel-based representations, 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.

Similar to these methods, the proposed approaches in this dissertation can delineate contours of actions and interactions. Hence, our output is more precise than cuboids. In Chapter 4 we propose an efficient approach that requires significantly fewer evaluations for localizing actions. We achieve this by learning the relations between the background and foreground action supervoxels. Supervoxels help to reduce the search space with fewer regions of interest in the testing videos. Furthermore, our proposed approach generates fewer but class-dependent hypotheses (or candidate locations), and the hypotheses for each action are the result of context walk where new observations depend on past observations.

Context has been used extensively for object detection [7, 14, 24]. Heitz and Koller [24] reduce the number of false positives using context between background and objects. Similarly, Alexe *et al.* [7] use context for object detection in images by learning relations betweens windows in training images to the ground truth bounding boxes. There are several works that use context for action recognition using different significations of the word ‘context’. Gupta and Davis [21] attempted to understand the relationship between actions and the objects used in those actions. Han *et al.* [22] also used object context for action recognition . However, both the methods assume that detectors for multiple objects are available. Ikizler-Cinbis and Sclaroff [29] used a variety of features associated with objects, actions and scenes to perform action recognition. They also required person detection using [19]. Marszalek *et al.* [54] used movie scripts as automatic supervision for scene and action recognition in movies. Zhang *et al.* [115] extracted motion words and utilized the relative locations between the motion words and a reference point in local regions to establish the spatio-temporal context for action recognition. Sun *et al.* [85] presented a hierarchical structure to model the context information of SIFT points, and their model consists of point-level, intra-trajectory, and inter-trajectory relationships. Wu *et al.* [104] incorporated context through spatio-temporal coordinates for action recognition.

Our proposed approach in Chapter 4 is inspired from Alexe *et al.* [7], and differs in several key aspects: (1) for precise detection of actions in videos, we cannot use windows or cuboids which can contain significant amounts of background due to articulation, actor/camera movement and naturally from cyclic actions. Furthermore, due to inherent differences between images and videos and the extra degree of freedom due to time, we segment the video into *supervoxels* to reduce the search space of candidate hypotheses. (2) Instead of pixels in images, our proposed approach operates on a *graph* where nodes represent supervoxels. (3) Since, we localize actions using supervoxels instead of 3D windows, we have to infer action locations using a Conditional Random Field on the graph created for the testing video. *In summary, ours is the first work that explicitly relies on both foreground actions and background for action localization with an emphasis on fewer number of classifier evaluations.*

2.1.2 Online Action Prediction

Online prediction aims to predict actions from partially observed videos *without* any localization. Prediction is a multi-action classification, where methods typically focus on maximum use of temporal, sequential and past information to predict labels and their confidences. Li and Fu [47] predict human activities by mining sequence patterns, and modeling causal relationships between them. Zhao *et al.* [116] represent the structure of streaming skeletons (poses) by a combination of human-body-part movements and use it to recognize actions in RGB-D. Hoai and De la Torre [25] simulate the sequential arrival of data while training, and train detectors to recognize incomplete events. Similarly, Lan *et al.* [45] propose hierarchical 'movemes' to describe human movements and develop a max-margin learning framework for future action prediction. Ryoo [73] proposed integral and dynamic bag-of-words for activity prediction, and divide the training and testing videos into small segments and match the segments sequentially. In follow-up work, Ryoo and Aggarwal [74] treat interacting people as a group and recognize interactions in continuous videos by computing group motion similarities. Similarly, Kong *et al.* [43] proposed to model temporal dynamics of human actions by explicitly considering all the history of observed features as well as features in smaller temporal segments. Yu *et al.* [113] predict actions using Spatial-Temporal

Implicit Shape Model (STISM), which characterizes the space-time structure of the sparse local features extracted from a video. Cao *et al.* [10] perform action prediction by applying sparse coding to derive the activity likelihood at small temporal segments, and later combine the likelihoods for all the segments. For the case of interactions, Huang and Katani [28] predict the reaction in a two-person setting by modeling it as an optimal control problem. Recently, there have been works on online temporal detection [15, 48] without localization.

In contrast to these works, in Chapter 5 we perform both action prediction and localization in an online manner in a streaming video, where action localization helps in action prediction and vice versa.

2.1.2.1 Pose for Recognition

Low-level motion features, both hand-crafted [94] and deep learned [13, 96], have imparted significant gains to the performance of action recognition and localization algorithms. However, human actions inherently consists of articulation which low-level features cannot model explicitly. The compact and low-dimensional nature of high-level representations such as human poses might make them sensitive and unstable for the task of action localization and recognition. Nonetheless, human pose estimation has been successfully employed for action recognition in several works. For instance, Majiwa *et al.* [53] implicitly capture poses through 'poselet activation vector' and later use them for action recognition in static images. Xu *et al.* [107] detect poses through [108] and couple them with independently computed local motion features around the joints for action recognition. Wang *et al.* [93] also extended [108] to videos and represented videos in terms of spatio-temporal configurations of joints to perform action recognition. Raptis and Cigal [68] recognize and detect interactions from videos by modeling poselets as latent variables in a structural SVM formulation. Joint recognition of action and pose estimation in videos was recently proposed by Xiaohan *et al.* [105]. They divide the action into poses and their spatio-temporal parts, and model their inter-relationships through And-Or graphs. Pirsiavash *et al.* [67] predict quality of sports actions by training a regression model from spatio-temporal pose features, to scores from expert judges. Poses were recently used for *offline* action localization by Wang *et al.* [95], who detect actions using a unified approach that discovers action parts using dynamical poselets, and the relations between them. Pose-based Convolutional Neural Network descriptor (P-CNN) [13] was used for the task of action recognition, and the authors concluded that correct estimation of human poses leads to significant improvements in action recognition.

Similarly, several works model and determine head orientation and upper body pose for recognition and localization of interactions. Patron-Perez *et al.* [62] developed a per-person descriptor which incorporates head orientation and the local spatio-temporal context around each person to detect interactions. Vahdat *et al.* [91] represented each individual by a set of key poses and formulated spatio-temporal relationships among them in their model. The frame-wise interaction model in Patron-Perez *et al.* [61] combines local and global descriptors and incorporates visual attention of people by modeling their head orientations. Although Hoai and Zisserman [26] do not detect poses per se, they develop a technique to detect different upper body configurations each consisting of multiple parts.

In contrast to these methods, in Chapter 5, we use pose in conjunction with low-level features and mid-level superpixels to predict and localize actions (interactions) in an online manner. Our work is at the cross roads of both online prediction and offline localization, in a unified framework for both actions and interactions operable in partially observed videos.

2.2 Unsupervised Action Analysis

Action localization so far in the literature has been handled in a *supervised* manner. Therefore, in Chapter 6 we are the first to propose an *unsupervised* action localization approach. This approach involves action clustering, annotation, and localization. There has been some recent work on *Unsupervised* action clustering, which aims at grouping videos of similar human actions into separate action classes *without* any action localization. These approaches use local features to compute similarity among action videos. Wang et al. [98] extracts the coarse shape of human figures to match pairs of action images using a linear programming approach. Savarese *et al.* [77] propose spatio-temporal correlograms to encode temporal information into motion features, which are used in an unsupervised generative model to learn action classes. Niebles et al. [57] use pLSA and LDA to learn intermediate topics associated with actions to cluster them. Yang et al. [109] discover sub-actions as motion primitives to construct a string matching similarity matrix for clustering. Jones and Shao [36] present a Feature Grouped Spectral Multigraph (FGSM) approach, that uses a spectral embedding on a feature graph to cluster actions. Liu *et al.* [49] use a hierarchical clustering multi-task learning method for jointly grouping and recognizing human actions. Jones and Shao [37] propose a Dual Assignment k-Means (DAKM) approach, which considers the contextual relations between actions and scenes for human action clustering.

In contrast, in Chapter 6, 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.

CHAPTER 3: HUMAN ACTION DATASET

Benchmark datasets are important to objectively measure the performance of proposed action recognition methods and establish a common ground for fair comparision with existing approaches. The majority of action recognition datasets suffer from two disadvantages: (1) The number of their classes is typically very low compared to the richness of performed actions by humans in reality, e.g. KTH [78], Weizmann [8], UCF Sports [71], IXMAS [100] datasets includes only 6, 9, 9, 11 classes respectively. (2) The videos are recorded in unrealistically controlled environments. For instance, KTH, Weizmann, IXMAS are staged by actors; HOHA [54] and UCF Sports are composed of movie clips captured by professional filming crew. Recently, web videos have been used in order to utilize unconstrained user-uploaded data to alleviate the second issue [44, 50, 56, 69]. However, the first disadvantage remains unresolved as the largest existing dataset does not include more than 51 actions while several works showed that the number of classes play a crucial role in evaluating an action recognition method [35, 69]. Therefore, we have compiled a new dataset with 101 actions and 13,320 clips which is nearly twice bigger than the largest existing dataset in terms of number of actions and clips. HMDB51 [44] and UCF50 [69] were the largest ones with 6,766 clips of 51 actions and 6,681 clips of 50 actions, respectively, at the time of release of our dataset.

The UCF101 dataset is composed of web videos which are recorded in unconstrained environments and typically include camera motion, various lighting conditions, partial occlusion, low quality frames. Fig. 3.1 shows sample frames of 6 action classes from UCF101.

Figure 3.1: Sample frames for 6 action classes of UCF101.

3.1 Dataset Details

Action Classes: UCF101 includes total number of 101 action classes which we have divided into five types: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, Sports.

UCF101 is an extension of UCF50 which included the following 50 action classes: {*Baseball Pitch, Basketball Shooting, Bench Press, Biking, Billiards Shot, Breaststroke, Clean and Jerk, Diving, Drumming*, *Fencing, Golf Swing, High Jump, Horse Race, Horse Riding, Hula Hoop*, *Javelin Throw,*, *Juggling Balls*, *Jumping Jack*, *Jump Rope*, *Kayaking, Lunges*, *Military Parade*, *Mixing Batter*, *Nun chucks*, *Pizza Tossing*, *Playing Guitar*, *Playing Piano*, *Playing Tabla*, *Playing Violin*, *Pole Vault, Pommel Horse, Pull Ups*, *Punch, Push Ups*, *Rock Climbing Indoor*, *Rope Climbing*, *Rowing, Salsa Spins*, *Skate Boarding*, *Skiing, Skijet, Soccer Juggling*, *Swing*, *TaiChi*, *Tennis Swing*, *Throw Discus*, *Trampoline Jumping*, *Volleyball Spiking, Walking with a dog*, *Yo Yo*}. The color class labels specify which predefined action type they belong to.

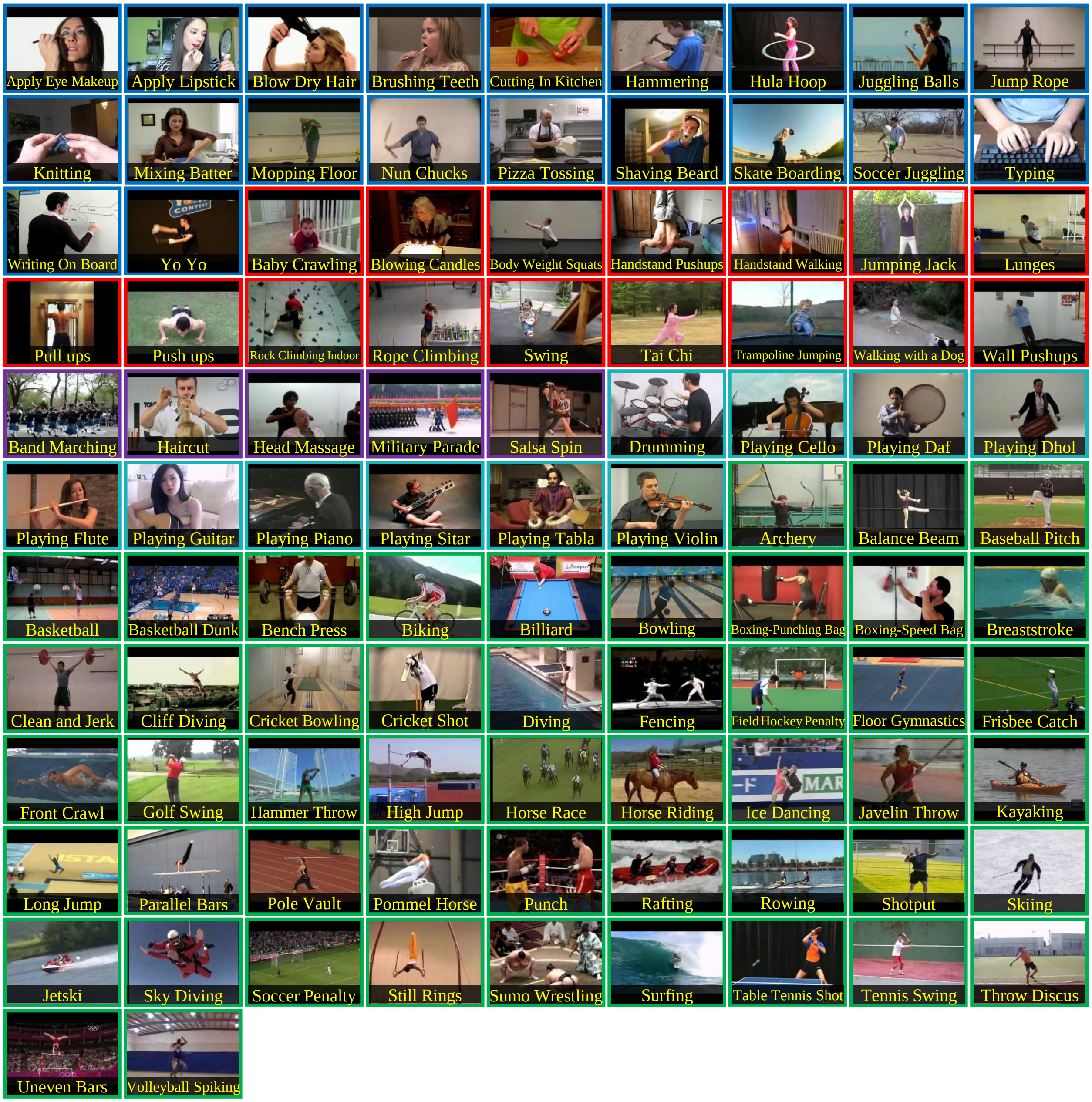


Figure 3.2: 101 actions included in UCF101 shown with one sample frame. The color of frame borders specifies to which action type they belong: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, Sports.

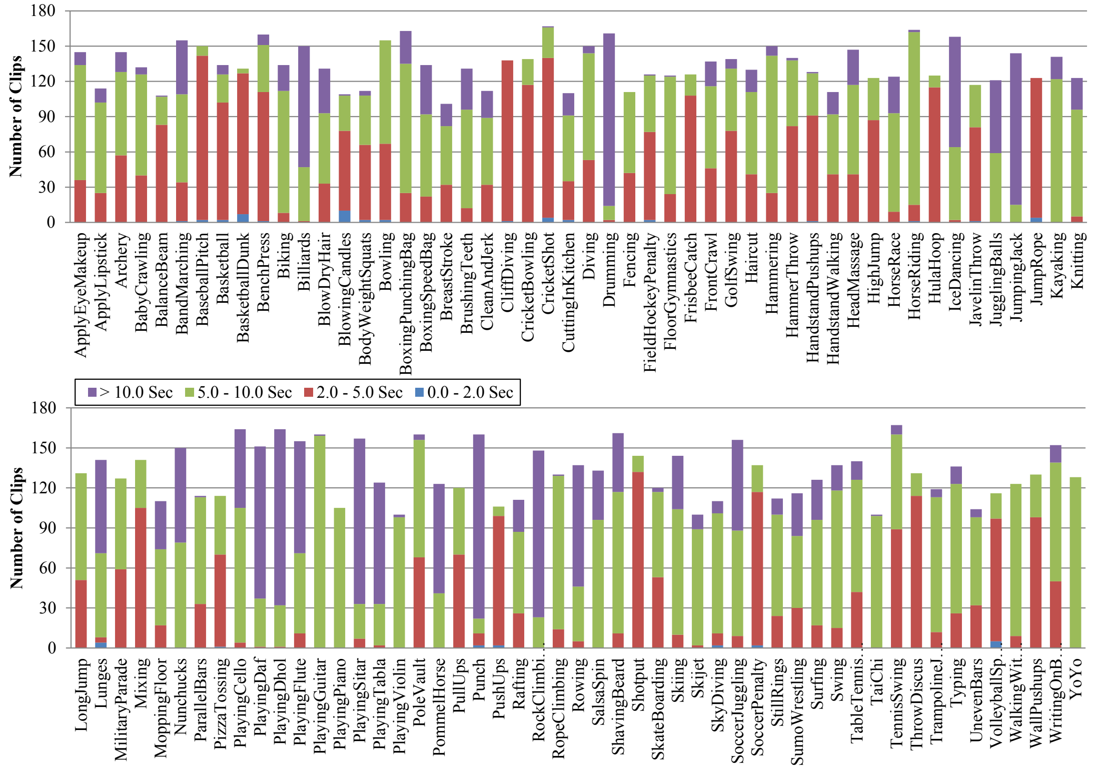


Figure 3.3: Number of clips per action class. The distribution of clip durations is illustrated by the colors.

The following 51 new classes are introduced in UCF101: {*Apply Eye Makeup*, *Apply Lipstick*, *Archery*, *Baby Crawling*, *Balance Beam*, *Band Marching*, *Basketball Dunk*, *Blow Drying Hair*, *Blowing Candles*, *Body Weight Squats*, *Bowling*,*Boxing-Punching Bag*, *Boxing-Speed Bag*, *Brushing Teeth*, *Cliff Diving*, *Cricket Bowling*, *Cricket Shot*, *Cutting In Kitchen*, *Field Hockey Penalty*, *Floor Gymnastics*, *Frisbee Catch*, *Front Crawl*, *Hair cut*, *Hammering*, *Hammer Throw*, *Handstand Pushups*, *Handstand Walking*, *Head Massage*, *Ice Dancing*, *Knitting*, *Long Jump*, *Mopping Floor*, *Parallel Bars*, *Playing Cello*, *Playing Daf*, *Playing Dhol*, *Playing Flute*, *Playing Sitar*, *Rafting*, *Shaving Beard*, *Shot put*, *Sky Diving*, *Soccer Penalty*, *Still Rings*, *Sumo Wrestling*, *Surfing*, *Table Tennis Shot*, *Typing*, *Uneven Bars*, *Wall Pushups*, *Writing On Board*}. Fig. 3.2 shows a sample frame for each action class of UCF101.

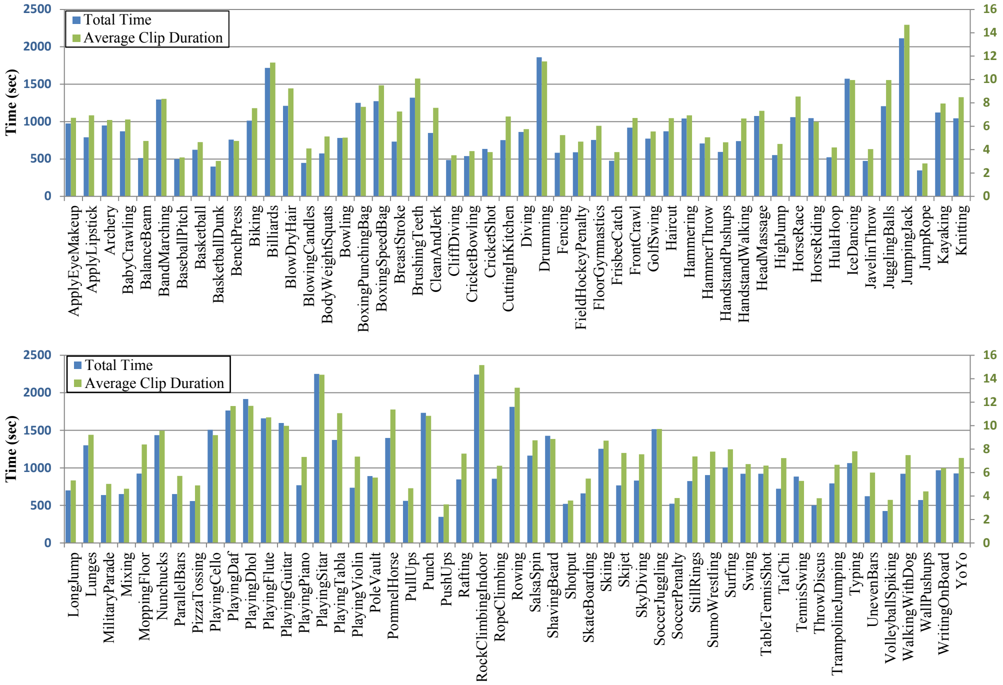


Figure 3.4: Total time of videos for each class is illustrated using the blue bars. The average length of the clips for each action is depicted in green.

Clip Groups: The clips of one action class are divided into 25 groups which contain 4-7 clips each. The clips in one group share some common features, such as the background or actors.

The bar chart of Fig. 3.3 shows the number of clips in each class. The colors on each bar illustrate the durations of different clips included in that class. The chart shown in Fig. 3.4 illustrates the average clip length (green) and total duration of clips (blue) for each action class.

The videos are downloaded from YouTube [3] and the irrelevant ones are manually removed. All clips have fixed frame rate and resolution of 25 FPS and 320×240 respectively. The videos are saved in `.avi` files compressed using *DivX* codec available in k-lite package [1]. The audio is preserved for the clips of the new 51 actions. Table 3.1 summarizes the characteristics of the dataset.

Table 3.1: Summary of Characteristics of UCF101

Actions	101
Clips	13320
Groups per Action	25
Clips per Group	4-7
Mean Clip Length	7.21 sec
Total Duration	1600 mins
Min Clip Length	1.06 sec
Max Clip Length	71.04 sec
Frame Rate	25 fps
Resolution	320×240
Audio	Yes (51 actions)

Naming Convention: The zipped file of the dataset (available at `http://crcv.ucf.edu/data/UCF101.php`) includes 101 folders each containing the clips of one action class. The name of each clip has the following form:

v_**X**_g**Y**_c**Z**.avi

where **X**, **Y** and **Z** represent action class label, group and clip number respectively. For instance, `v_ApplyEyeMakeup_g03_c04.avi` corresponds to the clip 4 of group 3 of action class `ApplyEyeMakeup`.

3.2 Experimental Results

We performed an experiment using bag of words approach which is widely accepted as a standard action recognition method to provide baseline results on UCF101.

From each clip, we extracted Harris3D corners (using the implementation by [54]) and computed

162 dimensional HOG/HOF descriptors for each. We clustered a randomly selected set of 100,000 space-time interest points (STIP) using k-means to build the codebook. The size of our codebook is k=4000 which is shown to yield good results over a wide range of datasets. The descriptors were assigned to their closest video words using nearest neighbor classifier, and each clip was represented by a 4000-dimensional histogram of its words. Utilizing *three train/test splits*, a SVM was trained using the histogram vectors of the training set. We employed a nonlinear multiclass SVM with histogram intersection kernel and 101 classes each representing one action. For testing, a similar histogram representation for the query video was computed and classified using the trained SVM. This method yielded an overall accuracy of 43.9%; The confusion matrix for all 101 actions is shown in Fig. 3.5.

The accuracy for the predefined action types are: Sports (49.40%), Playing Musical Instrument (42.04%), Human-Object Interaction (36.62%), Body-Motion Only (37.64%), Human-Human Interaction (42.66%). Sports actions achieve the highest accuracy since performing sports typically requires distinctive motions which makes the classification easier. Moreover, the background in sports clips are generally less cluttered compared to other action types. Unlike Sports Actions, Human-Object Interaction clips typically have a highly cluttered background. Additionally, the informative motions typically occupy a small portion of the motions in the clips which explains the low recognition accuracy of this action class.

We recommend a *three train/test split* (available at: `http://crcv.ucf.edu/data/UCF101/UCF101TrainTestSplits-RecognitionTask.zip`) experimental setup to keep consistency of the reported tests on UCF101; the baseline results provided in this section were computed using the same scenario. These train/test splits have been designed in a way to keep the groups separate, hence not sharing the clips from the same group in training and testing, as the clips within a group are obtained from a single long video. Each test split has 7 different groups and their respective remaining 18 groups are used for training.

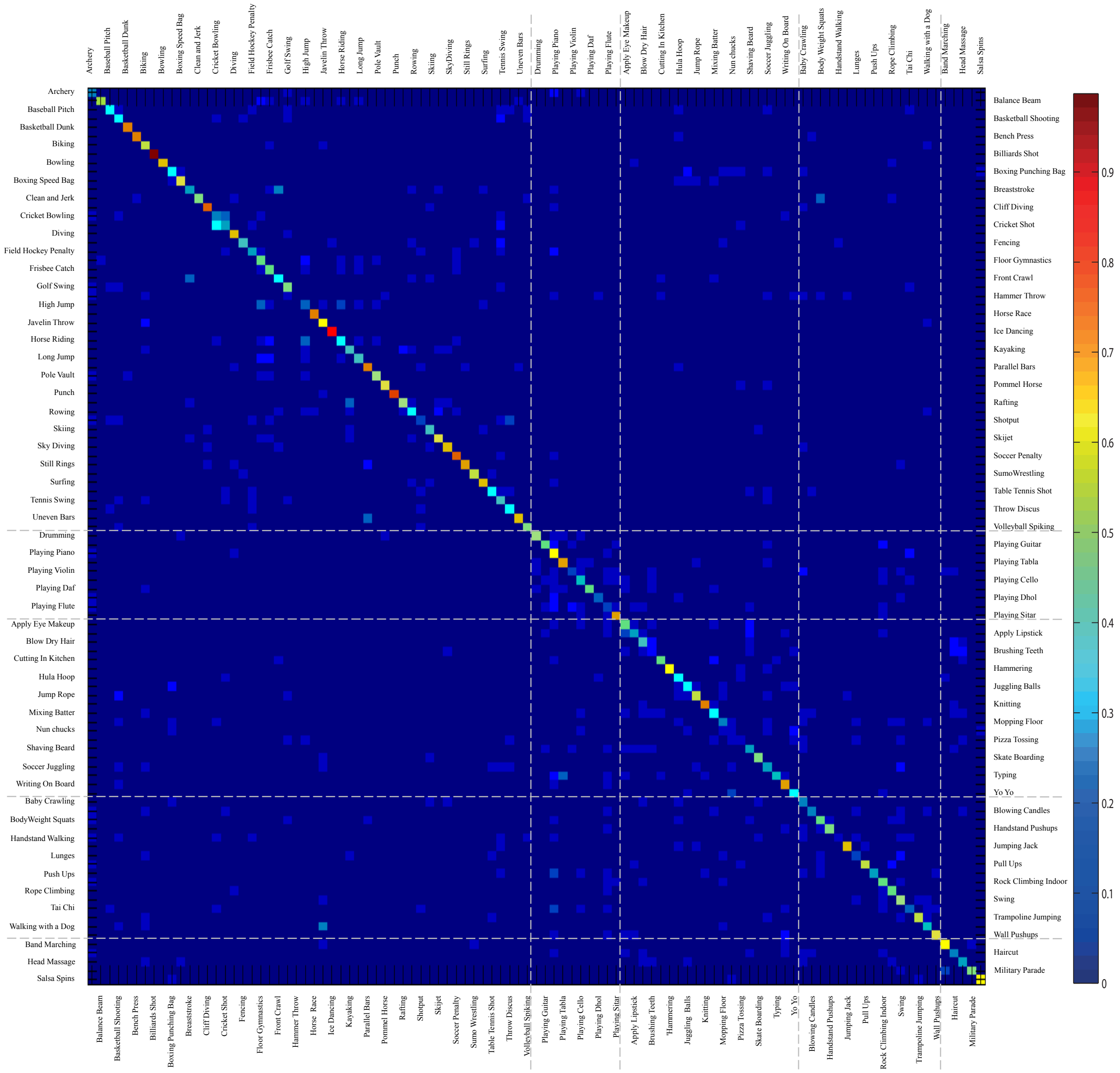


Figure 3.5: Confusion table of baseline action recognition results using bag of words (BOW) approach on UCF101. The drawn lines separate different types of actions; 1-50: Sports, 51-60: Playing Musical Instrument, 61-80: Human-Object Interaction, 81-96: Body-Motion Only, 97-101: Human-Human Interaction.

Table 3.2: Summary of Major Action Recognition Datasets

Dataset	Number of Actions	Clips	Background	Camera Motion	Release Year	Resource
KTH [78]	6	600	Static	Slight	2004	Actor Staged
Weizmann [8]	9	81	Static	No	2005	Actor Staged
UCF Sports [71]	9	182	Dynamic	Yes	2009	TV, Movies
IXMAS [100]	11	165	Static	No	2006	Actor Staged
UCF11 [50]	11	1168	Dynamic	Yes	2009	YouTube
HOHA [54]	12	2517	Dynamic	Yes	2009	Movies
Olympic [56]	16	800	Dynamic	Yes	2010	YouTube
UCF50 [69]	50	6681	Dynamic	Yes	2010	YouTube
HMDB51 [44]	51	6766	Dynamic	Yes	2011	Movies, YouTube, Web
UCF101	**101**	**13320**	Dynamic	Yes	**2012**	YouTube

The above experiment was also performed using a leave-one-group-out 25-fold cross validation setup, giving an overall accuracy of 44.5%. By testing on one group and training on the rest, it was made sure that the clips from a group are not divided between training and testing set.

3.3 Related Datasets

UCF Sports, UCF11, UCF50 and UCF101 are the four action datasets compiled by UCF in chronological order; each one includes its precursor. We made two minor modifications in the portion of UCF101 which includes UCF50 videos: the number of groups is fixed to 25 for all the actions, and each group includes up to 7 clips. Table 3.2 shows a list of existing action recognition datasets with detailed characteristics of each. Note that UCF101 is remarkably larger than the rest.

3.4 Summary

We introduced UCF101, the most challenging dataset for action recognition compared to the existing ones. It includes 101 action classes and over 13K clips which makes it outstandingly larger than other datasets. UCF101 is composed of unconstrained videos downloaded from YouTube which feature challenges such as poor lighting, cluttered background and severe camera motion. We provided baseline action recognition results on this new dataset using standard BOW method with overall accuracy of 43.9%.

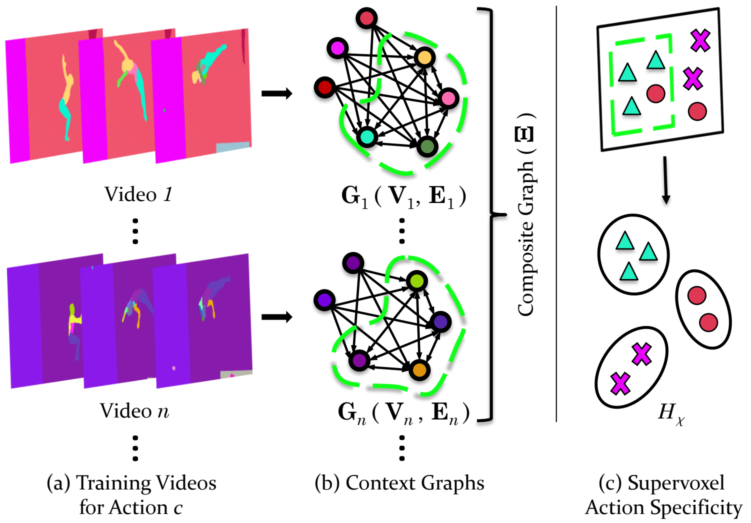
CHAPTER 4: ACTION LOCALIZATION THROUGH CONTEXT WALK

An action detector is typically learned on trimmed training videos, and during testing it is used to exhaustively search through a video to localize an action, using a sliding window approach. The search space spans the spatial dimensions of the frame and the temporal length of the video, where the spatio-temporal action volume is scaled in each dimension to try all possible combinations. Precise action localization also requires searching various tube (spatio-temporal volume) deformations to accommodate human articulation during the action. In realistic videos (e.g. from surveillance cameras), having longer duration and higher resolution, it becomes impractical to use sliding window approach to look for actions and interesting events. We propose an efficient action localization approach that learns contextual relations in the form of relative locations between different video regions. These learnt relations are used to perform a *context walk* in testing video to reduce the search space and evaluate the classifier at significantly fewer locations.

In this chapter, we describe in detail our approach for action localization in videos. The proposed approach begins by over-segmenting the training videos into supervoxels and computing the local features in the videos. For each training video, a graph is constructed that captures relations from all the supervoxels to those belonging to action foreground (ground truth) (see Fig. 4.1). Then, given a testing video, we initialize the context walk with a randomly selected supervoxel and find its nearest neighbors using appearance and motion features. The displacement relations from training supervoxels are then used to predict the location of an action in the testing video. This gives a conditional distribution for each supervoxel in the video of belonging to the action. By selecting the supervoxel with the highest probability, we make predictions about location of the action again and update the distribution. This *context walk* is executed for several steps and is followed by inferring the action proposals through CRF. The confidences for the localized action segments (proposals) are then obtained through Support Vector Machine learned using the labeled training videos (see Fig. 4.2).

4.1 Context Graphs for Training Videos

Let the index of training videos for action $c = 1 \dots C$ range between $n = 1 \dots N_c$, where N_c is number of training videos for action c. The ith supervoxel in the nth video is represented by $\mathbf{u}_n^i, i = 1 \dots I_n$, where I_n is the number of supervoxels in video n. Each supervoxel either belongs to a foreground action or the background. Next, we construct a directed graph $\mathbf{G}_n(\mathbf{V}_n, \mathbf{E}_n)$ for each training video across all the action classes. The nodes in the graph are represented by the supervoxels while edges $\mathbf{e}^{ij}$ emanate from all the nodes (supervoxels) to those belonging to the foreground, i.e., supervoxels spatio-temporally contained within the ground truth tube.


Video *1*
$\mathbf{G}_1(\mathbf{V}_1, \mathbf{E}_1)$
Video *n*
$\mathbf{G}_n(\mathbf{V}_n, \mathbf{E}_n)$
Composite Graph (Ξ)
H_χ

(a) Training Videos for Action *c*

(b) Context Graphs

(c) Supervoxel Action Specificity

Figure 4.1: This figure illustrates the idea of using context in the form of spatio-temporal displacements between supervoxels. (a) Given N_c videos for an action c which have been over-segmented into supervoxels, we construct a context graph for each video as shown in (b). Each graph has edges emanating from all the supervoxels to those that belong to foreground action (circumscribed with dashed green contours). The color of each node in (b) is the same as that of the corresponding supervoxel in (a). Finally, a composite graph (Ξ) from all the context graphs is constructed, implemented efficiently using a kd-tree. (c) We also quantify 'supervoxel action specificity' which returns the likelihood of a particular supervoxel belonging to an action and use it in conjunction with context to localize actions.

Let each supervoxel $\mathbf{u}$ be represented by its spatio-temporal centroid, i.e., $\mathbf{u}_n^i = (x_n^i, y_n^i, t_n^i)$. The features associated with $\mathbf{u}_n^i$ are given by $\mathbf{\Phi}_n^i = ({}_1\phi_n^i, {}_2\phi_n^i, \ldots, {}_F\phi_n^i)$, where F is the total number of features. Then, for a particular action c, the graphs $\mathbf{G}_n$ and features $\mathbf{\Phi}_n^i, \forall n = 1 \ldots N_c$ are represented by the composite graph Ξ_c which constitutes all the training information necessary to localize an action during testing. The following treatment is developed for each action class, therefore, we drop the subscript c for clarity and use it when necessary.

4.2 Context Walk in the Testing Video

For a testing video, we obtain supervoxels ($\sim 200 - 300$ per video) with each supervoxel and its local features represented by $\mathbf{v}$ and $\mathbf{\Phi}$, respectively. Then, we construct an undirected graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$ where $\mathbf{V}$ contains the supervoxels represented with spatio-temporal centroids, and $\mathbf{E}$ contains edges between neighboring supervoxels. Our goal is to find a contiguous subsets of nodes in this graph that form action proposals. We achieve this by making sequential observations based on context. Given the composite graph Ξ, we traverse the supervoxels in testing video in a sequence, referred to as *context walk*. The sequence till step $\tau \leq T$ is given by $\mathbf{S}_{\mathbf{v}}^{\tau} = (\mathbf{v}^1, \mathbf{v}^2, \ldots \mathbf{v}^{\tau})$, and $\mathbf{S}_{\mathbf{\Phi}}^{\tau} = (\mathbf{\Phi}^1, \mathbf{\Phi}^2, \ldots, \mathbf{\Phi}^{\tau})$. Each observed supervoxel during the walk independently proposes candidate supervoxels which are later visited if they accumulate enough support during the course of the walk. Next, we describe the procedure of generating such a sequence on a given testing video.

The initial supervoxel $\mathbf{v}^1$ is selected randomly in the testing video. We find similar supervoxels from the training data and project their displacement vectors to the selected supervoxel $\mathbf{v}^{\tau}$ in the testing video. The following function $\psi(.)$ with the associated parameters $\mathbf{w}_{\psi}$ generates a conditional distribution over all the supervoxels in the testing video given only the current supervoxel $\mathbf{v}^{\tau}$, its features $\mathbf{\Phi}^{\tau}$, and the composite graph Ξ, i.e.,

$$\psi(\mathbf{v}|\mathbf{v}^\tau, \mathbf{\Phi}^\tau, \Xi; \mathbf{w}_\psi) = Z^{-1} \sum_{n=1}^{N_c} \sum_{i=1}^{I_n} \sum_{j|e_{ij} \in \mathbf{E}_n} H_\sigma(\mathbf{\Phi}^\tau, \mathbf{\Phi}_n^i; \mathbf{w}_\sigma) \cdot H_\delta(\mathbf{v}, \mathbf{v}^\tau, \mathbf{u}_n^i, \mathbf{u}_n^j; w_\delta), \tag{4.1}$$

where H_σ computes the similarity between features of current supervoxel in testing video $\mathbf{\Phi}^\tau$, and all the training supervoxels ($\mathbf{\Phi}_n^i$). H_δ transfers displacements between supervoxels in training videos to a supervoxel in the testing video. Both functions have weight parameters $\mathbf{w}_\sigma$ and w_δ, respectively, and Z is the normalization factor. Theoretically, Eq. 4.1 loops over all displacement vectors in all the training videos, and is computationally prohibitive. Therefore, we only consider the nearest neighbors for the selected supervoxel during testing using kd-trees (one per action). In Eq. 4.1, the function H_δ assigns a confidence to each supervoxel $\mathbf{v}$ in the testing video whether it is part of the action or not. This is achieved by computing proximity of a supervoxel in the testing video to the displacement vector projected onto the current supervoxel $\mathbf{v}^\tau$. If $\mathbf{u}_n^j - \mathbf{u}_n^i$ defines the displacement vector from the supervoxel $\mathbf{u}_n^i$ to the foreground action supervoxel $\mathbf{u}_n^j$, then H_δ is given by:

$$H_\delta(\mathbf{v}, \mathbf{v}^\tau, \mathbf{u}_n^i, \mathbf{u}_n^j; w_\delta) = \exp\Big(- w_\delta \| \mathbf{v} - \big(\mathbf{v}^\tau + \mathbf{u}_n^j - \mathbf{u}_n^i\big) \| \Big). \tag{4.2}$$

Furthermore, the function H_σ in Eq. 4.1 is simply the weighted sum of distances between the different features:

$$H_\sigma(\mathbf{\Phi}^\tau, \mathbf{\Phi}_n^i; \mathbf{w}_\sigma) = \exp\Big(- \sum_{f=1}^{F} \Big(w_{\sigma_f} \Gamma_{\sigma_f}\big({}_f\phi^\tau, {}_f\phi_n^i\big)\Big)\Big), \tag{4.3}$$

where Γ_{σ_f} with the associated weight parameter w_{σ_f} defines the distance function for the fth feature. For the proposed method, we used the following features: (i) ${}_1\phi = (x, y, t, s)$, i.e., centroid of the supervoxel in addition to scale (or volume) s with each dimension normalized between 0 and 1 relative to the video, (ii) appearance and motion descriptor ${}_2\phi = \mathbf{d}$ using improved Dense Trajectory Features (iDTF) [94], and (iii) the supervoxel action specificity measure, as described in Sec. 4.2.1.

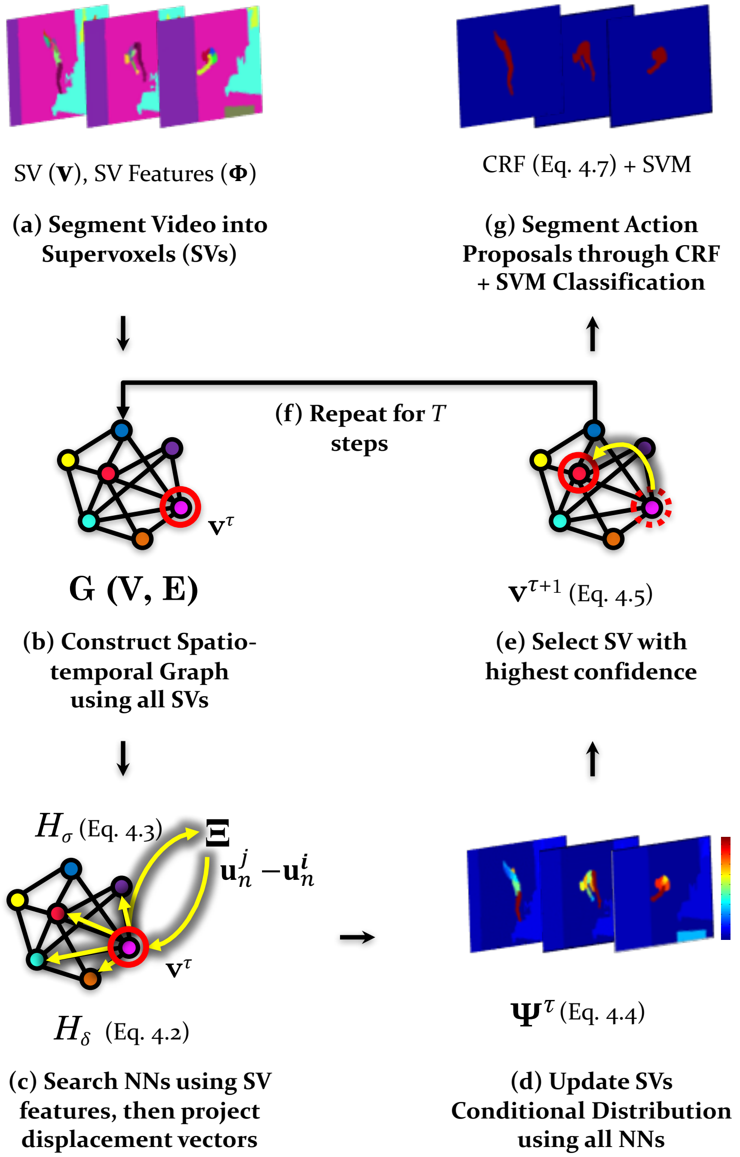

SV ($\mathbf{V}$), SV Features ($\mathbf{\Phi}$)
(a) Segment Video into Supervoxels (SVs)
CRF (Eq. 4.7) + SVM
(g) Segment Action Proposals through CRF + SVM Classification
(f) Repeat for T steps
$\mathbf{v}^{\tau}$
G (V, E)
(b) Construct Spatio-temporal Graph using all SVs
$\mathbf{v}^{\tau+1}$ (Eq. 4.5)
(e) Select SV with highest confidence
H_{σ} (Eq. 4.3)
Ξ
$\mathbf{u}_n^j - \mathbf{u}_n^i$
$\mathbf{v}^{\tau}$
H_{δ} (Eq. 4.2)
(c) Search NNs using SV features, then project displacement vectors
$\mathbf{\Psi}^{\tau}$ (Eq. 4.4)
(d) Update SVs Conditional Distribution using all NNs

Figure 4.2: This figure depicts the testing procedure of the proposed approach. (a) Given a testing video, we perform supervoxel (SV) segmentation. (b) A graph $\mathbf{G}$ is constructed using the supervoxels as nodes. (c) We find the nearest neighbors of the selected supervoxel ($\mathbf{v}^{\tau}$; initially selected randomly) in the composite graph Ξ which returns the displacement vectors learned during training. The displacement vectors are projected in the testing video as shown with yellow arrows. (d) We update the foreground/action confidences of all supervoxels using all the NNs and their displacement vectors. (e) The supervoxel with the highest confidence is selected as $\mathbf{v}^{\tau+1}$. (f) The walk is repeated for T steps. (g) Finally, a CRF gives action proposals whose action confidences are computed using SVM.

At each step τ, we compute the non-parametric conditional distribution $\psi(.)$ in Eq. 4.1 and use it to update $\Psi(.)$ in the following equation, which integrates the confidences that supervoxels gather during the context walk:

$$\Psi^{\tau}(\mathbf{v}|\mathbf{S}_{\mathbf{v}}^{\tau}, \mathbf{S}_{\boldsymbol{\Phi}}^{\tau}, \Xi; \mathbf{w}) = w_{\alpha}\psi(\mathbf{v}|\mathbf{v}^{\tau}, \boldsymbol{\Phi}^{\tau}, \Xi; \mathbf{w}_{\psi}) + (1 - w_{\alpha})\Psi^{\tau-1}(\mathbf{v}|\mathbf{S}_{\mathbf{v}}^{\tau-1}, \mathbf{S}_{\boldsymbol{\Phi}}^{\tau-1}, \Xi; \mathbf{w}), \quad (4.4)$$

where $\mathbf{w}$ are the parameters associated with Ψ. In the above equation, the conditional distribution Ψ is updated with exponential decay at the rate w_{α}. Finally, the supervoxel with the highest probability from Eq. 4.4 is selected to be visited in the next step of the context walk:

$$\mathbf{v}^{\tau+1} = \arg\max_{\mathbf{v}} \Psi^{\tau}(\mathbf{v}|\mathbf{S}_{\mathbf{v}}^{\tau}, \mathbf{S}_{\boldsymbol{\Phi}}^{\tau}, \Xi; \mathbf{w}). \quad (4.5)$$

Each video typically contains several hundred supervoxels. Although kd-tree significantly speeds up the Eq. 4.1, the efficiency of nearest neighbor search can be further improved using feature compression techniques [58].

4.2.1 Measuring Supervoxel Action Specificity

In a testing video, some supervoxels are distinct and discriminative towards one action while other supervoxels might be discriminative for other actions. We quantify this observation using a simple technique where we cluster all the descriptors (iDTF [94]) from the training videos of a particular action c into $k_c = 1 \ldots K$ clusters. Our goal is to give each supervoxel an action specificity score. Let $\xi(k_c)$ represent the ratio of number of supervoxels from foreground (ground truth) of action c in cluster k_c to all the supervoxels from action c in that cluster. Then, given the appearance/motion descriptors $\mathbf{d}$, if the supervoxel belongs to cluster k_c, its action specificity $H_{\chi}(\mathbf{v}^i)$ is quantified as:

$$H_{\chi}(\mathbf{v}^i) = \xi(k_c) \cdot \exp\Big(\frac{\|\mathbf{d}^i - \mathbf{d}_{k_c}\|}{r_{k_c}}\Big), \tag{4.6}$$

where $\mathbf{d}_{k_c}$ and r_{k_c} are the center and radius for the kth cluster, respectively.

4.2.2 Inferring Action Locations using 3D-CRF

Once we have the conditional distribution $\mathbf{\Psi}^T(.)$, we merge the supervoxels belonging to actions so that resulting action proposals have contiguous supervoxels without any gaps or voids. For this, we use a Conditional Random Field where nodes form the supervoxels while edges link neighboring supervoxels. We minimize the negative log-likelihood over all supervoxel labels $\mathbf{a}$ in the video:

$$-\log\big(Pr(\mathbf{a}|\mathbf{G}, \mathbf{\Phi}, \mathbf{\Psi}^T; w_{\Upsilon})\big) = \sum_{\mathbf{v}^i \in \mathbf{V}} \Big(\Theta\big(a^i|\mathbf{v}^i, \mathbf{\Psi}^T\big) + \sum_{\mathbf{v}^j|\mathbf{e}^{ij} \in \mathbf{E}} \Upsilon\big(a^i, a^j|\mathbf{v}^i, \mathbf{v}^j, \mathbf{\Phi}^i, \mathbf{\Phi}^j; w_{\Upsilon}\big)\Big),$$

where v^i and v^j are the neighboring supervoxel nodes in the undirected graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$, connected by the edge e^{ij}, and a^i and a^j are their respective supervoxel labels. w_{Υ} is the associated parameter with the function $\Upsilon(.)$. $\Theta(.)$ captures the unary potential and depends on the conditional distribution in Eq. 4.4 after T steps and action specificity measure computed through Eq. 4.6, both of which are normalized between 0 and 1:

$$\Theta\big(a^i|\mathbf{v}^i, \mathbf{\Psi}^T\big) = -\log\Big(H_{\chi}(\mathbf{v}^i) \cdot \mathbf{\Psi}^T(\mathbf{v}^i)\Big). \tag{4.7}$$

If Ω^i and Ω^j are the volumes of the ith and jth supervoxel, respectively, then the binary potential $\Upsilon(.)$ between neighboring supervoxels with parameter w_{Υ} is given by:

$$\Upsilon\big(a^i, a^j|\mathbf{v}^i, \mathbf{v}^j, \mathbf{\Phi}^i, \mathbf{\Phi}^j; w_{\Upsilon}\big) = w_{\Upsilon}\Gamma_d(\mathbf{d}^i, \mathbf{d}^j)\Big(|\log(\Omega^i/\Omega^j)| + |\Omega^i - \Omega^j|\Big), \tag{4.8}$$

where $\Gamma_d(.)$ measures the feature similarity between neighboring supervoxels.

Once we have the actions segmented in the testing video, we use Support Vector Machine to obtain the confidence for each action segment using the appearance/motion descriptors of all supervoxels in each segment.

4.3 Experiments

We evaluate the proposed approach on three challenging action localization datasets: UCF-Sports [71], sub-JHMDB [32, 95] and THUMOS'13 [33]. First, we provide experimental details about the three datasets followed by detailed analysis of the performance and complexity of the proposed algorithm.

Experimental Setup: For each video in the training and testing data, we obtain a supervoxel based segmentation using [59]. This is followed by extraction of improved Dense Trajectory Features (iDTF: HOG, HOF, MBH, Traj) [94]. Every supervoxel in the video is encoded using bag-of-words (BoW) representation on iDTFs. For all our experiments, we use Top-20 nearest neighbors using kd-trees with context walk executed for $T = 5$ steps. Once we obtain segments using CRF, an SVM with histogram-intersection kernel is used to classify each action segment. We train a one-vs-all SVM per action class using ground truth bounding boxes from training videos as positive samples, while negative samples are randomly selected from the background and other action classes. Each sample is a supervoxel based BoW histogram and we consider supervoxels as positive samples only if they overlap ($\geq 80\%$) with the ground truth. Features from all the supervoxels within the ground truth are accumulated to form one representative descriptor for SVM training. Furthermore, since we used normalized features, the parameters for $\psi(.)$ did not require tuning and were set to 1, i.e., $w_\delta = w_{\sigma_1} = w_{\sigma_2} = w_{\sigma_3} = 1$. The decay rate was set to $w_\alpha = 0.5$ and the weight for CRF was set to $w_\Upsilon = 0.1$ using training data.

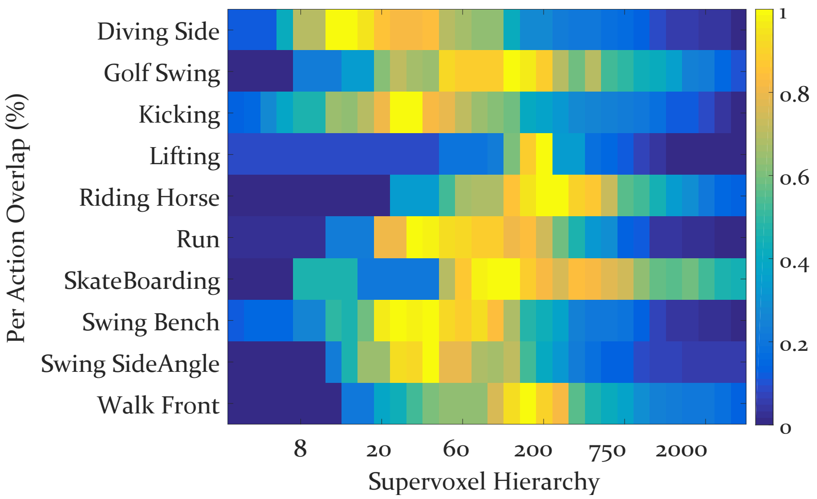


Figure 4.3: This figure shows the average of maximum supervoxel overlap in every training video of different actions as a function of segmentation level. Using the correct level from the hierarchy reduces the number of potential supervoxels we have to handle while testing. This speeds up the method without sacrificing performance.

Selecting Segmentation Level: Supervoxel methods [30, 59] generate different levels of a segmentation hierarchy. Each level has a different number of segmented supervoxels and may or may not cover an action. Searching for an action over the entire hierarchy is computationally inefficient and can also significantly hurt the performance of localization if an incorrect level in the hierarchy is selected. Manually choosing the correct level for a dataset is cumbersome since every action has its own complexity characterized by variation in scale, background clutter, and actor/camera movement. To automatically choose the right hierarchy level, we sample training videos from each action, and within every level of the hierarchy we find the overlap of the supervoxels with the ground truth bounding boxes. We take the maximum supervoxel overlap for each video and average it for all training videos of an action at a particular level of the segmentation hierarchy. Fig. 4.3 shows the average of maximum supervoxel overlap for each action at different levels of the hierarchy. The overlap peaks at a certain level and reduces thereafter. The average maximum supervoxel overlap varies for every action and selecting a unique level of segmentation for each action using this technique helps in correctly localizing an action in testing videos.

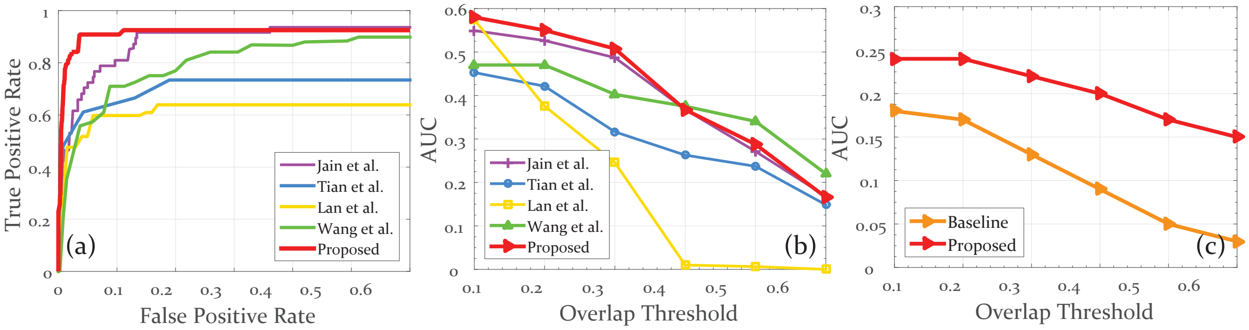


Figure 4.4: The ROC and AUC curves on UCF Sports Dataset [71] are shown in (a) and (b), respectively. The results are shown for Jain *et al.* [30] (orchid), Tian *et al.* [86] (blue), Lan *et al.* [46] (amber), Wang *et al.* [95] (green) and Proposed Method (red). (c) shows the AUC for THUMOS'13 dataset [33], for which we are the first to report results.

4.3.1 Experiments on UCF-Sports

The UCF Sports dataset [71] consists of 150 videos collected from broadcast television channels. The dataset includes 10 action classes: *diving, golf swing, kicking, etc*. Videos in the dataset are captured in a realistic setting with intra-class variations, camera motion, background clutter, scale and viewpoint changes. We follow evaluation methodology of Lan *et al.* [46] using the same train-test splits with intersection-over-union criterion at an overlap of 20%.

We construct a codebook ($K = 1000$) of iDTFs [94] using all the training videos. The quantitative comparison with state-of-the-art methods using ROC and Area Under Curve (AUC) for overlaps of $10\%, 20\%, 30\%, 40\%, 50\%$ and 60% is shown in Fig. 4.4(a,b). The ROC curve highlights that the proposed method performs better than the state-of-the-art methods [30, 46, 86, 95]. Although, we evaluated the classifier on very few segments of supervoxels, we are still able to achieve better results at an overlap of 20%. The comparison using AUC measure (Fig. 4.4(b)) also shows that we are able to achieve comparable results for different overlaps. We accredit this level of performance to avoiding background clutter and irrelevant camera motion through the use of context which allows the proposed method to ignore the potential false positive regions in the videos.

4.3.2 Experiments on THUMOS'13

THUMOS'13 action localization dataset was released as part of the THUMOS Challenge workshop [33] in 2013. This dataset is a subset of UCF-101 and has 3207 videos with 24 action classes such as *basketball, biking, cliff diving, etc*. The dataset is quite challenging and is currently the largest dataset for action localization. It includes several complex interactive actions such as *salsa spin, fencing, cricket bowling* with multiple action instances in the same video. We are the first to report action localization results on THUMOS'13. We also evaluated a competitive baseline using iDTFs with BoW ($K = 4000$), and trained a one-vs-all SVM-HIK for each action. Given a test video, we perform an exhaustive multi-scale spatio-temporal sub-volume search. The results are shown in Fig. 4.4(c).

4.3.3 Experiments on sub-JHMDB

The sub-JHMDB dataset [95] is a subset of the JHMDB [32] dataset where all the joints for humans in the videos have been annotated. Similar to [95], we use the box encompassing the joints as the ground truth. This dataset contains 316 clips over 12 action classes: *catch, climb stairs, golf, etc*. Jhuang *et al.* [32] have shown that this subset is far more challenging in recognizing actions compared to the entire dataset. The probable reason is the presence of the entire human body which exhibits complex variations in appearance and motion.

We used $K = 4000$ codebook centers for bag-of-words representation of the supervoxels. We report our results in Fig. 4.5 using both ROC and AUC curves. At an overlap of 20%, we perform better than the state-of-the-art and achieve competitive results at other overlapping thresholds.

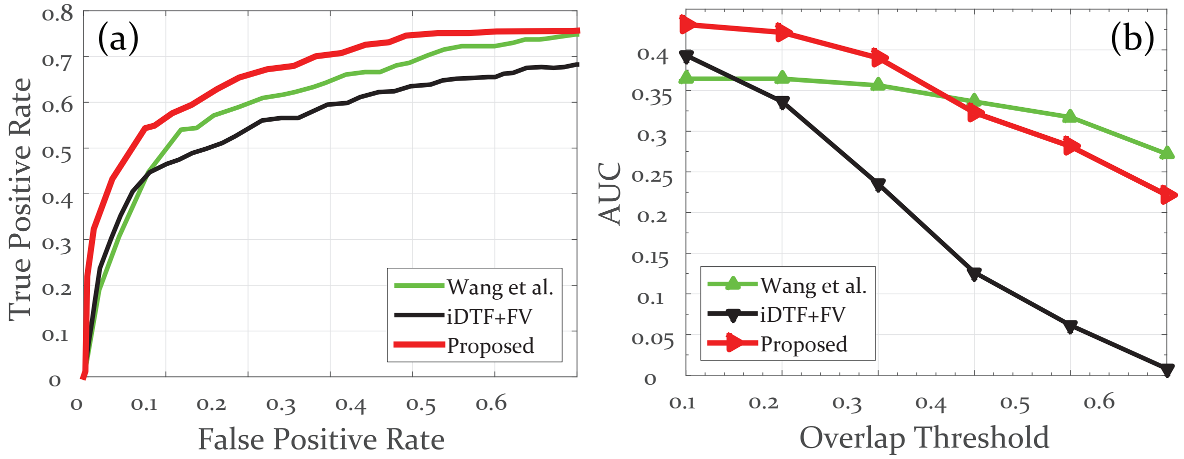


Figure 4.5: The ROC and AUC curves for sub-JHMDB dataset [32, 95] are shown in (a) and (b), respectively. Green and black curves are from the method by Wang *et al.* [95] and their iDTF + Fisher Vector baseline. Red curve shows the performance of the proposed method which is better than [95].

Note that Wang *et al.* [95] also evaluated a competitive baseline over this dataset. This baseline uses iDTF features with a Fisher Vector encoding (black curves in Fig. 4.5) to exhaustively scan at various spatio-temporal locations at multiple scales in the video. Performing better than the baseline in a far more efficient manner emphasizes the strength of the proposed approach and reinforces that context does make a significant impact in understanding and predicting the locations of actions.

4.3.4 Analysis and Discussion

In Table 4.1, we report the percentage AUC on UCF-Sports [71] and sub-JHMDB [95] datasets. These numbers are computed at an overlap of 20% and show we perform competitively or better than existing approaches.

Table 4.1: Quantitative comparison of proposed approach with existing methods at 20% overlap.

Method	UCF-Sports	sub-JHMDB
Wang *et al.* [95]	47%	36%
Wang *et al.* (iDTF+FV) [95]	-	34%
Jain *et al.* [30]	53%	-
Tian *et al.* [86]	42%	-
Lan *et al.* [46]	38%	-
Proposed	**55%**	**42%**

Computational Efficiency: Our approach achieves competitive results compared to the state-of-the-art methods on multiple datasets. However, in certain cases, some existing methods show better accuracy at higher overlaps, but this does come with a price of evaluating classifiers at a significantly higher number of locations. Note that, the BOW framework in our approach is only a matter of choice and efficiency, and results are expected to improve further through Fisher Vectors [60, 95].

Component's Contributions: The proposed approach has several aspects that contribute to its performance. We quantify their relative contributions to overall performance in Fig. 4.6, which shows both the ROC and AUC curves computed on UCF-Sports dataset. The grey curves represent the output using just supervoxel action specificity (§4.2.1). Here, we assign confidences using Eq. 4.6 to each supervoxel, followed by a fixed threshold. Each segment is considered as an action segment and evaluated using the ground truth. Next, we incorporate context walk as shown with green curves in Fig. 4.6. In this case, the confidence for supervoxels are obtained using Eq. 5.10. The difference between grey and red curves highlights the importance of context for action localization. Next, we show improvement in performance obtained by using CRF (Eq. 5.9) in blue curves, which helps in obtaining contiguous and complete action segments. Finally, the performance obtained with all aspects of the proposed approach (including SVM) is shown with red curves. The reason SVM gives a large boost is that the evaluation of action localization simultaneously quantifies action classification. Correctly localizing the action but assigning it an incorrect label is treated as incorrect localization. Since each SVM is trained on both background and negative samples from other classes, it significantly contributes to the correct classification of the localized actions. Note that for non-linear kernels, the summation of scores from supervoxels does not equal that of an action volume, thus, necessitating classification using an SVM. Nevertheless, this is an inexpensive step since we require very few SVM evaluations.

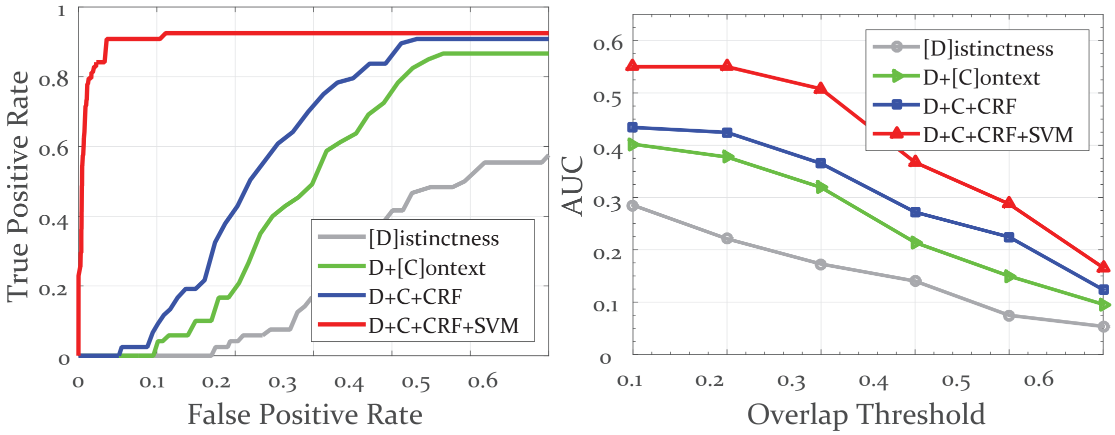


Figure 4.6: This figure shows the contributions of the fours aspects of the proposed approach towards overall performance, in terms of ROC (left) and AUC of Precision-Recall curve as a function of overlap threshold (right).

Action Contours: The proposed approach uses over-segmented supervoxels, therefore, it produces action segments which can be used for video segmentation as well. Since the local features (iDTF) are based on motion, the segments are heavily dependent on the motion of actors. Such results are an improvement over cuboid representation which can contain significant quantities of background. Some qualitative results of the proposed approach with segmented actors are presented in Fig. 4.7.

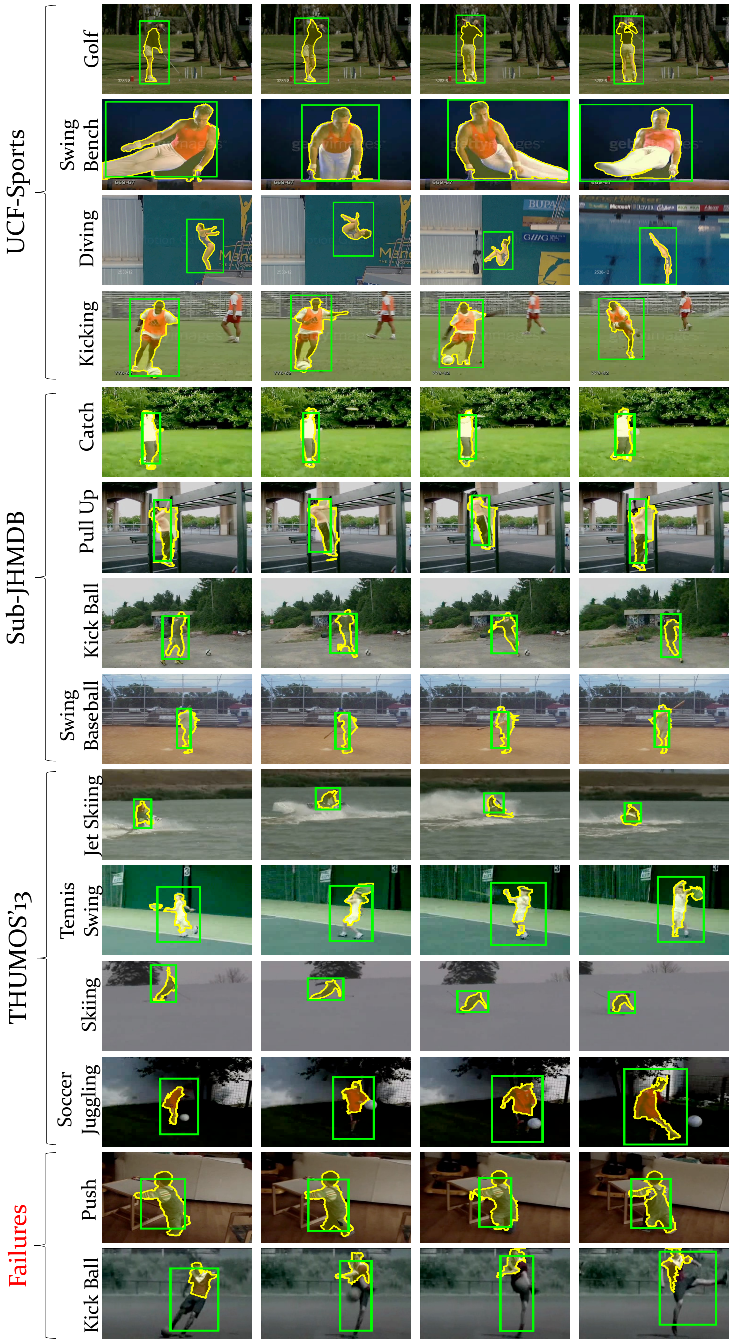


Figure 4.7: This figure shows qualitative results of the proposed approach (yellow contours) against ground truth (green boxes) on selected frames of testing videos. The first four rows are from UCF-Sports [71], next four are from sub-JHMDB [95], and then the next four rows are from THUMOS'13 [33] datasets. Last two rows show two failure cases from sub-JHMDB dataset.

Since the proposed method uses supervoxels to segment the video, we are able to capture the entire human body contours after CRF. These results show that supervoxels indeed help in obtaining fine contours while reducing the complexity of the problem. However, there are certain cases where the proposed approach fails as shown in the last two rows of Fig. 4.7. The action depicted on the second last row shows the case where the action *push* was classified as *walk*, even though it was localized correctly. The set of images on the last row shows incorrect localization of the action *kickball*. For this particular case, the large motion of the actor resulted in a large number of supervoxels on the lower body as compared to training videos. Many supervoxels had different distances (from Eq. 4.2) as compared to the ones seen during training. This caused lower confidences for such supervoxels resulting in only upper-body localization.

Complexity Analysis: We offer an analysis of the number of classifier evaluations of the proposed approach on the number of supervoxels or subvolumes with two other state-of-the-art methods. Table 4.2 shows Tian *et al.* [86] who learn a Spatio-temporal Deformable Parts Model detector that is used to search for an action over width (X), height (Y), time (T) and different aspect ratios (S) within the video. This requires enormous computations which can incur many false positives as well. We also compare the effectiveness of the proposed approach to Jain *et al.* [30], who also use supervoxels to reduce computation. Given N supervoxels at the lowest level, they apply an agglomerative hierarchical clustering, which merges supervoxels at each level of the hierarchy followed by an application of SVM classifier on each supervoxel. Compared to these approaches we localize in constant time (context-walk with 5 steps and one inference through CRF followed by an execution of SVM). Note that this table only shows the complexity of localizing the action, assuming the features have been computed and models have been learnt in advance.

Table 4.2: Number of classifier evaluations as a function of supervoxels / subvolumes in a video.

Method	Evaluated Volumes	Complexity
SDPM [86]	*XYTS*	$\mathcal{O}(n^4)$
Action Tubelets [30]	*N* + *(N-1)* + . . . + *1*	$\mathcal{O}(n^2)$
Proposed	5 (+ CRF)	$\mathcal{O}(c)$

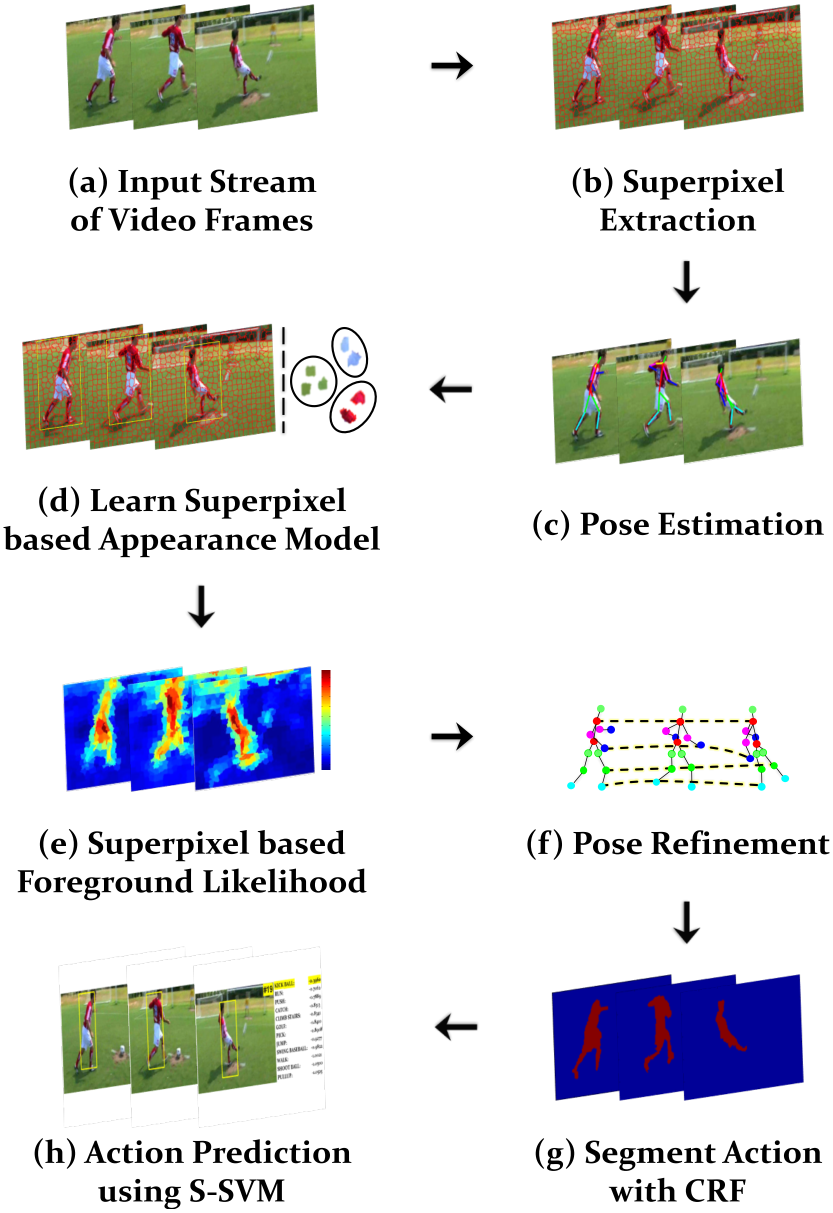
4.4 Summary

We presented an efficient and effective approach to localize actions in videos. We use context to make a series of observations on supervoxels, such that the probability of predicting the location of an action increases at each step. Starting with a random supervoxel, we find similar supervoxels from the training data, and transfer the knowledge about relative spatio-temporal location of an action to the test video. This gives a conditional distribution over the graph formed by supervoxels in the testing video. After selecting the supervoxel with highest probability, we repeat the steps. The conditional distribution at the end of Context Walk over supervoxel graph is used in a CRF to infer the number and location of action proposals. Finally, each of the proposals is evaluated through an SVM. Due to both supervoxels and context, the proposed approach requires very few classifier evaluations. In the next chapter, we address the limitations of *offline* action localization approaches by addressing a new problem of *online* action localization in streaming videos.

CHAPTER 5: ONLINE LOCALIZATION AND PREDICTION OF ACTIONS AND INTERACTIONS

Visual surveillance and human-machine interaction applications require timely localization of activities. In the event of an abnormal action of a criminal nature, an immediate automated response or alert can be helpful to notify the authorities. Similarly, a pedestrian walking in front of an autonomously driven car, can detect the human action (*walking*) to alter its path or come to a halt. *Offline* approaches, as presented in Chapter 4, have the entire video at their disposal, to first detect and then recognize the activity. Therefore, they can observe the entire motion of an activity and search through a video to localize it. This makes them inapplicable for the above mentioned real-life problems. Hence, in this chapter we propose to solve this new problem of *Online Action and Interaction Localization*. In comparison, an *online* approach would use partially observed video, and rely on limited motion information to detect the action in every frame, as well as predict the action that possibly occurs in the near future.

This chapter focuses on two related problems of: 1) Online Localization and 2) Online Prediction of Actions and Interactions. We begin by describing the technical details of *Online Localization*, which uses Superpixel-based and Pose-based foreground likelihoods to infer location of the actor using Conditional Random Field. We then describe two proposed approaches for *Online Prediction*: (1) Binary-SVM with dynamic programming hybrid and (2) Structural SVM based approach. Next, we perform our experimental evaluation on six challenging action and interaction datasets. Finally, we discuss and analyze the variance of localization and prediction performance as a function of time.



(a) Input Stream of Video Frames

(b) Superpixel Extraction

(c) Pose Estimation

(d) Learn Superpixel based Appearance Model

(e) Superpixel based Foreground Likelihood

(f) Pose Refinement

(g) Segment Action with CRF

(h) Action Prediction using S-SVM

Figure 5.1: This figure shows the framework of the approach proposed in this chapter. (a) Given an input video, (b) we over-segment each frame into superpixels and (c) detect poses using an off-the-shelf method [99]. (d) An appearance model is learned using all the superpixels inside a pose bounding box as positive, and those outside as negative samples. (e) In a new frame, the appearance model is applied on each superpixel of the frame to obtain a foreground likelihood. (f) To handle the issue of visual drift, poses are refined using spatio-temporal smoothness constraints on motion and appearance. (g) Finally, a CRF is used to obtain local action proposals at each frame, (h) on which actions (interactions) are predicted through Structural SVM.

5.1 Online Localization of Actions and Interactions

The pipeline of our approach for localization is shown in Fig. 5.1. Given a testing video, we initialize the localization algorithm with several pose estimations in individual frames and refine the poses using multiple spatio-temporal constraints from previous frames. Next, we segment the testing video frames into superpixels. The features computed within each superpixel are used to learn a superpixel-based appearance model, which distinguishes the foreground from the background by training a discriminative classifier with superpixels within each pose bounding box as foreground and the rest of superpixels as background. Simultaneously, the conditional probability of pose hypotheses at current time-step (frame) is computed using pose confidences and consistency with poses estimated in previous frames. The superpixel and pose-based foreground probability is used to infer the location of actors at each frame through a Conditional Random Field enforcing spatio-temporal smoothness in color, optical flow, motion boundaries and edges among superpixels. After localizing actions (interactions) at each time-step (frame), we refine poses by imposing consistency in locations and appearance of joints as well as scale of poses. Once the pose has been estimated and refined at current time-step, the superpixel-based appearance model is updated to avoid visual drift. This process is repeated for every frame in an online manner (see Fig. 5.1) and gives human localization at every frame. Note that, the pose refinement only aids in estimating pose for current frame, the poses of past frames remain unchanged as expected for an online approach.

After localization, the spatio-temporal tubes are then used for prediction and recognition of labels at each frame, discussed later in Sec. 5.2. Thus, the pose estimation not only provides initialization for the proposed discriminative appearance models, as it is more robust compared to human detection in action (interaction) videos due to articulation, it also allows computation of pose features which we use use during label prediction (Sec. 5.2). Note that the pose estimations can consist of any or multiple body configurations such as upper or full body, as well as multiple humans interacting or performing actions. To simplify the treatment in this section, we assume we are dealing with a single actor or action, where the case of multiple actors is handled by solving the correspondence problem first (e.g. using Hungarian Algorithm, in our case) followed by independent treatment of each actor for localization and action prediction.

Let $\mathbf{s}_t$ represent a superpixel by its centroid in frame t and $\mathbf{p}_t$ represent one of the poses in frame t. Since our goal is to localize the actor in each frame, we use $\mathbf{X}_t$ to represent, a sequence of bounding boxes (tube) in a small window of δ frames. Each bounding box is represented by its centroid, width and height. Similarly, let $\mathbf{S}_t$ and $\mathbf{P}_t$ respectively represent all the superpixels and poses at that time instant. Given the pose and superpixel-based observations till time t, $\mathbf{S}_{1:t}$ and $\mathbf{P}_{1:t}$, the state estimate $\mathbf{X}_t$ at time t is obtained using the following equation through Bayes Rule:

$$p(\mathbf{X}_t|\mathbf{S}_{1:t},\mathbf{P}_{1:t}) = Z^{-1}p(\mathbf{S}_t|\mathbf{X}_t).p(\mathbf{P}_t|\mathbf{X}_t).\int p(\mathbf{X}_t|\mathbf{X}_{t-1}).p(\mathbf{X}_{t-1}|\mathbf{S}_{1:t-1},\mathbf{P}_{1:t-1})d\mathbf{X}_{t-1}, \quad (5.1)$$

where Z is the normalization factor, and the state transition model is assumed to be Gaussian distributed, i.e., $p(\mathbf{X}_t|\mathbf{X}_{t-1}) = \mathcal{N}(\mathbf{X}_t;\mathbf{X}_{t-1},\mathbf{\Sigma})$. Eq. 5.1 accumulates the evidence over time on the superpixels and poses in streaming mode. The state which maximizes the posterior (MAP) estimate in Eq. 5.1 is selected as the new state. An implication of Eq. 5.1 is that the state or localization cannot be altered in the past frames, which makes online localization different from the existing offline methods. Next, we define the pose and superpixel-based foreground likelihoods used for estimating Eq. 5.1.

5.1.1 Superpixel-based Foreground Likelihood

Learning an appearance model helps in distinguishing the foreground actions (interactions) from the background. Given foreground and background superpixels in the previous frames $t-\delta,\ldots,t-1$, we group them into $k = 1\ldots K$ clusters. Furthermore, let ζ_k define the ratio of foreground to background superpixels for the kth cluster through k-means. Then, the appearance-based foreground score using color, ϕ_{color}, and flow, ϕ_{flow}, features in the superpixels is given by:

$$H_{\text{fg}}(\mathbf{s}_t) = \exp\Big(\frac{\|\boldsymbol{\phi}_{\text{color}}(\mathbf{s}_t) - \mathbf{q}_k)\|}{r_k}\Big) \cdot \zeta_k + \exp\Big(\frac{\|\boldsymbol{\phi}_{\text{flow}}(s_t) - \boldsymbol{\mu}_k\|}{\rho_k}\Big), \tag{5.2}$$

where $\mathbf{q}_k$ and r_k are the cluster center and radius, respectively, whereas $\boldsymbol{\mu}_k$ and ρ_k represent the mean and variance of optical flow for the kth cluster.

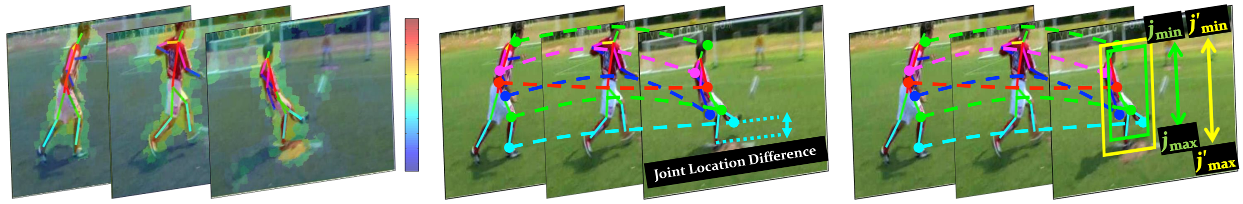
In Eq. 5.2, the clusters are updated incrementally at each time-step (frame) to recover from the visual drift using a temporal window of past δ frames. Note that, background superpixels (e.g. grass in Fig. 5.1) within a pose bounding box are inevitably considered as foreground initially, however the later segmentation through Conditional Random Field serves to alleviate this problem finding a fine actor contour and separating foreground superpixels within the pose bounding box. The ζ_k helps to compensate for this issue by quantifying the foreground/background ratio for each cluster. Finally, the superpixel-based foreground likelihood in Eq. 5.1 is given as: $p(\mathbf{S}_t|\mathbf{X}_t) = \alpha_{\text{fg}} \cdot H_{\text{fg}}(\mathbf{s}_t)$, where α_{fg} is the normalization factor.

5.1.2 Pose-based Foreground Likelihood

We represent each pose $\mathbf{p}_t$ graphically with a tree, given by $\mathbf{T} = (\boldsymbol{\Pi}, \boldsymbol{\Lambda})$. The body joints $\pi \in \boldsymbol{\Pi}$ are based on appearance connected by $\lambda \in \boldsymbol{\Lambda}$ edges capturing the structure of the pose. The joint j with its location in pose $\mathbf{p}_t$ is represented by $\boldsymbol{\pi}_t^j$, consisting of its x and y locations. Then, the raw cost (or negated detection score) for a particular pose $\mathbf{p}_t$ is the sum of appearance and deformation costs:

$$H_{\text{raw}}(\mathbf{p}_t) = \sum_{j \in \boldsymbol{\Pi}_t} \psi(\boldsymbol{\pi}_t^j) + \sum_{(j,j') \in \boldsymbol{\Lambda}_t} \chi(\boldsymbol{\pi}_t^j, \boldsymbol{\pi}_t^{j'}), \tag{5.3}$$

where ψ and χ are linear functions of appearance features of pose joints, and the relative joint displacements (deformations) w.r.t each other.



(a) (b) (c)

Figure 5.2: This figure shows a visualization of the joint smoothness costs used in pose-based foreground likelihood for (a) appearance smoothness of joints (J_{app}), (b) location smoothness of joints (J_{loc}) and (c) scale smoothness of joints (J_{sc}).

We use a pre-trained pose detector to obtain pose hypotheses in each frame. In [82], we used Flexible Mixture-of-Parts [108] for pose estimation, which optimizes over latent variables that capture different joint locations and pose configurations. In this chapter, we report results using Convolutional Pose Machines (CPM) [99] which uses deep learning. For CPM, the deformation costs are embedded within joint costs in Eq. 5.3. Since the pose estimation in both methods works on individual frames, it is inherently noisy and does not take into account the temporal information available in videos. We impose the following smoothness constraints (as shown in Fig. 5.2 (a-c)) in the previous δ frames to re-evaluate poses in Eq. 5.3 for the current time-step.

Appearance Smoothness of Joints: Since the appearance of a joint is not expected to change drastically in a short window of time, we impose the appearance consistency between superpixels at joint locations:

$$J_{\text{app}}(\mathbf{p}_t) = \sum_{j=1}^{|\mathbf{\Pi}_t|} \|H_{\text{fg}}(\hat{\mathbf{s}}_t^j) - H_{\text{fg}}(\hat{\mathbf{s}}_{t-1}^j)\|, \tag{5.4}$$

where $\hat{\mathbf{s}}_t^j$ is the enclosing superpixel of the joint $\boldsymbol{\pi}_t^j$.

Location Smoothness of Joints: Since human motion is naturally smooth, we ensure that displacements in joint locations over time are small. This is achieved by fitting a 2D spline using piecewise polynomials to each joint j on the past δ frames, $\boldsymbol{\gamma}_t^j$. Then the location smoothness cost over all joints is given by:

$$J_{\text{loc}}(\mathbf{p}_t) = \sum_{j=1}^{|\mathbf{\Pi}_t|} \|\boldsymbol{\gamma}_t^j - \boldsymbol{\pi}_t^j\|. \tag{5.5}$$

Scale Smoothness of Joints: Let j_{min} and j_{max} denote the vertical minimum and maximum for all the splines $\boldsymbol{\gamma}_\tau, \forall\tau \in \{t-\delta, \dots, t\}$, i.e., the y-axis components of the bounding box circumscribing all the splines fitted on joints. Furthermore, let $j'_{\text{min}}, j'_{\text{max}}$ denote minimum and maximum for joints in actual poses $\boldsymbol{\pi}_t \in \mathbf{\Pi}_t$. Then, the scale smoothness cost essentially computes the overlap between the two heights:

$$J_{\text{sc}}(\mathbf{p}_t) = \|(j_{\text{max}} - j_{\text{min}}) - (j'_{\text{max}} - j'_{\text{min}})\|. \tag{5.6}$$

The combined cost of a particular pose is defined as its raw cost plus the smoothness costs across space and time, i.e.,

$$H_{\text{pose}}(\mathbf{p}_t) = H_{\text{raw}}(\mathbf{p}_t) + J_{\text{app}}(\mathbf{p}_t) + J_{\text{loc}}(\mathbf{p}_t) + J_{\text{sc}}(\mathbf{p}_t). \tag{5.7}$$

The change in pose and appearance of an actor may cause visual drift. Similar to Sec. 5.1.1, we use a temporal window of past δ frames to refine the pose locations. This helps in better prediction of the highly probable foreground locations in current frame. We propose an iterative approach to select poses in the past $\{t-\delta, \dots, t\}$ frames. Given an initial set of poses, we fit a spline to each joint $\boldsymbol{\pi}_t^j$. Then, our goal is to select a set of poses from $t-\delta$ to t frames, such that the following cost function is minimized:

$$({}^*\mathbf{p}_{t-\delta}, \dots, {}^*\mathbf{p}_t) = \underset{\mathbf{p}_{t-\delta},\dots,\mathbf{p}_t}{\arg\min} \sum_{\tau=t-\delta}^{t} \Big(H_{\text{pose}}(\mathbf{p}_\tau) \Big). \tag{5.8}$$

Algorithm 1 : Algorithm to refine pose locations in a batch of frames in Q iterations.

Input: $\mathbf{P}_{t-\delta}, \ldots, \mathbf{P}_t$
Output: ${}^*\mathbf{p}_{t-\delta}, \ldots, {}^*\mathbf{p}_t$

```
procedure RefinePoses()
    for $\tau = t - \delta$ to $t$ do
        ${}^*\mathbf{p}_\tau = \arg\min(H_{\text{raw}}(\mathbf{p}_\tau))$
    end for
    for $n = 1$ to $Q$ do
        Fit a spline $\boldsymbol{\gamma}^j$ to each joint using locations
        $[{}^*\boldsymbol{\pi}^j_{t-\delta}, \ldots, {}^*\boldsymbol{\pi}^j_t]$
        Compute $J_{\text{app}}(\mathbf{p}_t)$ using Eq. 5.4
        Compute $J_{\text{loc}}(\mathbf{p}_t)$ using Eq. 5.5
        Compute $J_{\text{sc}}(\mathbf{p}_t)$ using Eq. 5.6
        Find $({}^*\mathbf{p}_{t-\delta}, \ldots, {}^*\mathbf{p}_t)$ through Eq. 5.8.
    end for
end procedure
```

This function optimizes over pose detection, and the appearance, location and scale smoothness costs of joints (see Fig.5.1 (e)) by greedily selecting the minimum cost pose in every frame through multiple iterations, such that the joints are spatially accurate and temporally consistent with the motion of the action. In case of occlusions, the joint locations are projected using a linear motion model into future frames. This procedure is summarized in Algorithm 1. Note that the poses in previous frames of the batch are only refined simultaneously, however, the pose at the current time-step is used by the algorithm. Finally, the pose-based foreground likelihood in Eq. 5.1 is given by $p(\mathbf{P}_t|\mathbf{X}_t) = \exp(\alpha_{\text{pose}} \cdot H_{\text{pose}}(\mathbf{p}_t))$, where α_{pose} is the normalization factor.

5.1.3 Actor Segmentation using Conditional Random Fields (CRF)

Once we have the superpixel and pose-based foreground likelihoods in Eq. 5.1, we proceed to infer the action segment and its contour using a history of δ frames. Although the action location is computed online for every frame, using past δ frames adds robustness to segmentation. We form a graph with superpixels as nodes connected through *spatial* and *temporal* edges. Let variable a denote the foreground/background label of a superpixel. Then, the objective function of CRF becomes:

$$
\begin{aligned}
-\log\big(p(a_{t-\delta},\dots,a_t|\mathbf{s}_{t-\delta},\dots,\mathbf{s}_t,\mathbf{p}_{t-\delta},\dots,\mathbf{p}_t)\big) = \sum_{\tau=t-\delta}^{t} \Big(\underbrace{\mathbf{\Theta}\big(a_\tau|\mathbf{s}_\tau,\mathbf{p}_\tau\big)}_{\text{unary potential}} + \underbrace{\mathbf{\Upsilon}\big(a_\tau,a'_\tau|\mathbf{s}_\tau,\mathbf{s}'_\tau\big)}_{\text{spatial smoothness}} \Big) \\
+ \sum_{\tau=t-\delta}^{t-1} \underbrace{\mathbf{\Gamma}\big(a_\tau,a'_{\tau+1}|\mathbf{s}_\tau,\mathbf{s}'_{\tau+1}\big)}_{\text{temporal smoothness}}, \quad (5.9)
\end{aligned}
$$

where s_τ is the superpixel and p_τ is the pose in frame τ. The unary potential, with the associated weights symbolized with α, is given by:

$$
\mathbf{\Theta}\big(a_\tau|\mathbf{s}_\tau,\mathbf{p}_\tau\big) = \alpha_{\text{fg}}H_{\text{fg}}(\mathbf{s}_\tau) + \alpha_{\text{pose}}H_{\text{pose}}(\mathbf{p}_\tau), \quad (5.10)
$$

and the spatial and temporal binary potentials, with weights β and distance functions d, are given by:

$$
\begin{aligned}
\mathbf{\Upsilon}\big(a_\tau,a'_\tau|\mathbf{s}_\tau,\mathbf{s}'_\tau\big) = \beta_{\text{col}}d_{\text{col}}(\mathbf{s}_\tau,\mathbf{s}'_\tau) + \beta_{\text{hof}}d_{\text{hof}}(\mathbf{s}_\tau,\mathbf{s}'_\tau) + \beta_\mu d_\mu(\mathbf{s}_\tau,\mathbf{s}'_\tau) \\
+ \beta_{\text{mb}}d_{\text{mb}}(\mathbf{s}_\tau,\mathbf{s}'_\tau) + \beta_{\text{edge}}d_{\text{edge}}(\mathbf{s}_\tau,\mathbf{s}'_\tau), \quad (5.11)
\end{aligned}
$$

and

$$
\mathbf{\Gamma}\big(a_\tau,a'_{\tau-1}|\mathbf{s}_\tau,\mathbf{s}'_{\tau-1}\big) = \beta_{\text{col}}d_{\text{col}}(\mathbf{s}_\tau,\mathbf{s}'_{\tau-1}) + \beta_{\text{hof}}d_{\text{hof}}(\mathbf{s}_\tau,\mathbf{s}'_{\tau-1}) + \beta_\mu d_\mu(\mathbf{s}_\tau,\mathbf{s}'_{\tau-1}), \quad (5.12)
$$

respectively. In Eqs. 5.11, and 5.12, $\beta_{\text{col}}d_{\text{col}}(.)$ is the cost of color features in HSI color space, $\beta_{\text{hof}}d_{\text{hof}}(.)$ and $\beta_\mu d_\mu(.)$ compute compatibility between histogram of optical flow and mean of optical flow magnitude of the two superpixels, respectively. Similarly, $\beta_{\text{mb}}d_{\text{mb}}(.)$ and $\beta_{\text{edge}}d_{\text{edge}}(.)$ quantify incompatibility between superpixels with prominent boundaries.

5.2 Online Prediction of Actions and Interactions

For online recognition and class-label prediction of actions (interactions) in streaming videos, the classifier has to be applied on-the-fly on short temporal intervals. In particular, training videos of an action (interaction) class c are divided into M clips of equal-sized interval Ω. The average segment for each action is saved as prior information, which during testing allows us to compute features in intervals of the desired length. Next, we present a baseline approach using Support Vector Machine and Dynamic Programming hybrid [82] which divides videos into short segments, and trains a classifier independently for each segment. The online update of action confidences is achieved through dynamic programming on segment scores. In this chapter, we present an alternate approach which makes structured prediction by training a single classifier per action and modeling temporal dependence between action segments. In this section, we present the formulation in terms of linear classifiers for simplicity.

Let m index over temporal segments, i.e., $m \in 1, \ldots, M$ and $\mathbf{x}_{i,m}$ denote the mth segment and its feature vector in video i. Next, we present the two approaches to recognize and predict the class label at time t of a testing video.

5.2.1 Binary SVMs with Dynamic Programming Inference (DP-SVM)

First, we present a baseline for online prediction [82] in our localization framework. For training binary SVMs for segments in an action (interaction) class c, we assume availability of N trimmed positive and negative training videos. For linear SVM, we optimize the following objective function,

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}_m\|^2 + C\sum_{i=1}^{N}\sum_{m=1}^{M}\xi_{i,m} \\ \text{s.t.} \quad & \mathbf{y}_{i,m}\langle \mathbf{w}_m, \mathbf{x}_{i,m}\rangle \geq 1-\xi_{i,m},\ \xi_{i,m}\geq 0, \qquad \forall i,m \end{aligned} \tag{5.13}$$

where ξ represents slack variables, $\mathbf{w}_m$ is the single weight vector obtained per segment, C controls the trade-off between regularizer and constraints, and $\mathbf{y}_{i,m} = 1$, for desired m if $i \in c$ and -1, otherwise. Effectively, the training videos are divided into short intervals and an SVM is trained for each interval m independently. While testing on videos, the classification is performed on features accumulated on interval lengths learned from training videos. To exploit and preserve the sequential information present in videos, this is followed by dynamic programming on the short interval clips. At each step of the dynamic programming, the system effectively searches for the best matching segment that maximizes the SVM confidences from past segments. This method is applied independently for each class, and gives the confidence for that class. This shares resemblance to Dynamic Bag-Of-Words [73] which used RBF kernel to compute score between training and testing segments, and applied it on trimmed videos.

Let $F(t,z)$ be the result of dynamic programming at time t assuming the current interval is z for a particular class. The result of applying classifier on testing video o on features computed between $t-\Omega$ and t is given by $\sigma(\langle \mathbf{w}_m, \mathbf{x}_{o,t}\rangle)$, where σ is the sigmoid function. If the testing video is trimmed, then $F(t,z)$ is computed using the following recursion:

$$F(t,z) = \max_{m} F(t-\Omega, z-m)\cdot\sigma(\langle \mathbf{w}_m, \mathbf{x}_{o,t}\rangle), \tag{5.14}$$

where m is the index for temporal segment. At each time instant, the maximum value at time t gives the desired confidence for the class under consideration.

5.2.2 Structural SVM (S-SVM)

Ideally the prediction confidence for the correct class should increase as more action (interaction) in the video is observed over time. There is rich structure that can be derived from division of actions into sub-actions, and modeling the spatio-temporal dependence between them. S-SVM gives the ability to model these sub-actions as structured predictions. It also simplifies learning by using a single S-SVM classifier for action prediction, instead of learning multiple Binary SVMs and applying Dynamic Programming to accumulate scores (as in Sec. 5.2.1). Given testing video segments, we then apply Structural SVM detector to each segment of the test video. For this case, we redefine intervals w.r.t start time of an action (interaction), i.e., the start time of interval m is 0 of the trimmed training video. We set the problem in a Structural Support Vector Machine (S-SVM) with margin re-scaling construction, given by:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N*M}\xi_i \\ \text{s.t.} \quad & \langle \mathbf{w}, \mathbf{\Psi}_i(\mathbf{x}_i,\mathbf{y}_i) - \mathbf{\Psi}_i(\mathbf{x}_i,\mathbf{y})\rangle \geq \mathbf{\Delta}(\mathbf{y}_i,\mathbf{y}) - \xi_i, \\ & \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i, \xi_i \geq 0, \forall i \end{aligned} \tag{5.15}$$

where the joint feature map for input and output is given by $\mathbf{\Psi}(\mathbf{x},\mathbf{y}) = \mathbf{x}\cdot\text{sign}(\mathbf{y})$, and $\mathcal{Y} = \{-1, 1, \ldots, M\}$ is the set of all labels . In Eq. 5.15, ξ represents the slack variables for the soft-margin SVM, which optimizes over the learned weight vector $\mathbf{w}$ and the slack variables ξ. The constraint with the loss function $\mathbf{\Delta}(\mathbf{y}_i,\mathbf{y})$ ensures that the score with the correct label $\mathbf{y}_i$ is greater than alternate labels. Since the number of constraints can be tremendous, only subset of constraints are used during the optimization. For each training sample, the label $\mathbf{y}$ which maximizes $\langle \mathbf{w}, \mathbf{\Psi}(\mathbf{x}_i,\mathbf{y})\rangle + \mathbf{\Delta}(\mathbf{y}_i,\mathbf{y})$ is found and the constraint which maximizes this loss is added into the subset, known as the most violated constraint. For both actions and interactions, the temporal component of the loss is defined as:

$$\boldsymbol{\Delta}(\mathbf{y}_i, \mathbf{y}_{i'}) = \begin{cases} |\mathbf{y}_i - \mathbf{y}_{i'}|, & i \in c \wedge i' \in c \\ M + \epsilon, & i \in c \wedge i' \notin c \\ \epsilon, & \text{otherwise.} \end{cases} \tag{5.16}$$

The above loss function ensures that the confidence increases as the action (interaction) happens in the testing video, i.e., the evaluation during a positive test instance, possibly over a long video, yields a unique signature of confidence values that increases over time. This approach results in one S-SVM per class, and can be applied indiscriminately to untrimmed videos. For interactions an additional loss captures the relationship between actors. Once the weight vector $\mathbf{w}$ has been learned, the score for a clip in the testing video is computed using $\arg\max_{\mathbf{y}\in\mathcal{Y}}\langle \mathbf{w}, \boldsymbol{\Psi}(\mathbf{x}, \mathbf{y})\rangle$.

Note that the performance of detection or prediction for action (interaction) localization depends on the quality of localized tubes / cuboids, as the classifiers are only evaluated on such video segments. This is in contrast to previous prediction methods [25, 43, 47, 73] which do not spatially localize the actions (interactions).

5.3 Experiments

We evaluate our *online action localization* approach on six challenging datasets: 1) JHMDB, 2) Sub-JHMDB, 3) MSR-II, 4) UCF Sports, 5) TV Human Interaction and 6) UT Interaction datasets. We provide details for the experimental setup followed by the performance evaluation and analysis of the proposed approach.

Features: For each frame of the testing video we extract superpixels using SLIC [5]. This is followed by extraction of color features (HSI) for each superpixel, as well as improved Dense Trajectory features (iDTF: HOG, HOF, MBH, Traj) [94] within the streamed volumes of the video. Each superpixel descriptor has a length of 512 and we set $K = 20$. The pose detections are obtained using [99] and pose features using [32]. We build a vocabulary of 20 words for each pose feature, and represent pose with a 180d vector.

Parameters and Distance Functions: We use Euclidean distance for d_μ, chi-squared distance for d_{hof} and d_{col}, and geodesic distance for d_{mb} and d_{edge}. We unit-normalize the histograms before computing distances in CRF, therefore, we set absolute values of all the parameters α and β to 1.

Evaluation Metrics: Since the online localization algorithm generates tubes or cuboids with associated confidences, the Receiver Operating Characteristic (ROC) curves are computed at fixed overlap thresholds with the GT tubes. Following experimental setup of [46], we show ROC @ 20% overlap. Furthermore, Area Under the Curve (AUC) of ROC at various thresholds gives an overall measure of performance. Mean Average Precision (mAP) is also reported as it is not sensitive to True Negatives unlike ROC. The proposed evaluation metrics are computed over all action and interaction datasets for consistency. For MSR-II dataset, we also report results using Precision and Recall curves typically used for this dataset.

Inspired from early action recognition and prediction works [73], we also quantify the performance as a function of *Observation Percentage* of actions (interactions). For this evaluation method, the localization and prediction for testing videos are sampled at different percentages of observed video/action $(0, 0.1, 0.2, \ldots, 1)$. The ROC curve is computed at multiple overlap thresholds, and AUC is computed under ROC curves at respective thresholds. Accuracy for the prediction task is akin to classification and recognition. For instance, computing accuracy at 20% observation percentage entails finding label which occurs most among all the frames between $0 - 20\%$ of the action video, followed by multi-label classification. In the case of untrimmed videos, evaluation of prediction accuracy is not straightforward. For that, we first report prediction accuracy as a function of observation percentage within the temporal boundaries of ground truth actions (interactions) - similar to the trimmed case. Second, we extract hundreds of clips from each video, some of which contain ground truth actions, whereas others represent the background part of the untrimmed video. Then, we compute prediction accuracy as a function of observation percentage over each clip. This measure captures the false positives and offers more holistic evaluation on untrimmed videos.

Note that, in online action (interaction) localization, the prediction and localization is performed instantaneously at each frame in a streaming video, therefore once locations are detected and predictions are made, retroactive modifications or changes to results are not possible.

5.3.1 Datasets

JHMDB Dataset: The JHMDB [32] dataset is a subset of the larger HMDB51 [44] dataset collected from digitized movies and YouTube videos. It contains 928 videos consisting of 21 action classes. The dataset has annotations for all the body joints and has recently been used for offline action localization [20]. We use a codebook size of $K = 4000$ to train SVMs using iDTF features.

sub-JHMDB Dataset: The sub-JHMDB dataset has all human body joints visible in each frame. It contains a total of 316 videos over 12 action classes: *catch, climb stairs, golf, kick ball,* etc. The presence of the entire human within each frame makes it more challenging to recognize and localize the actions as compared to JHMDB dataset, due to high articulation of human body and joints, and complex variations in appearance and motion compared to partial body actions [32]. A codebook size of $K = 4000$ was used for IDTF, and SVMs were trained with a bag-of-words representation inside the ground truth action volumes.

UCF-Sports Dataset: The UCF Sports [71, 83] dataset is collected from broadcast television channels and consists of 150 videos. It includes a total of 10 action classes: *diving, golf swing, kicking, lifting, riding horse, skateboarding,* etc. Videos are captured in a realistic setting with intra-class variations, camera motion, background clutter, scale and viewpoint changes. We evaluated our method using the methodology proposed by [46], who use a train-test split with intersection-over-union criterion at an overlap of 20%. To train SVM, we use a codebook size of $K = 1000$ on iDTFs using all the training videos.

MSR-II Dataset: The MSR-II dataset [114] consists of 54 untrimmed videos and 3 action classes: Boxing, Handclapping and Handwaving. We follow the experimental methodology of [114], having cross-dataset evaluation, where KTH [78] dataset is used for training and testing is performed on MSR-II dataset. A codebook size of $K = 1000$ was used to train SVM on iDTFs. We show quantitative comparison using Precision-Recall curves with state-of-the-art *offline* methods. However, for uniformity with other datasets we also report results using ROC and AUC curves.

TV Human Interaction (TVHI): The TVHI dataset [61, 62] is collected from 23 different TV shows and is composed of 300 untrimmed videos. It includes 4 interaction classes: *hand shake, high five, hug and kiss*, with 50 videos each. It also contains a negative class with 100 videos, that have none of the listed interactions. The videos have varying number of actors in each scene, different scales and abrupt changes in camera viewpoint at shot boundaries. For our experiments we only use the 4 interaction classes (excluding negative class) for interaction localization. We use the suggested experimental setup of two train/test splits. The localization performance is reported using ROC and AUC curves.

UT Interaction: The UT Interaction dataset [74, 75] contains untrimmed videos of 6 interaction classes: *hand-shaking, hugging, kicking, pointing, punching, and pushing.* Similar to [61], we add *being kicked, being punched* and *being pushed* as interactions.

H

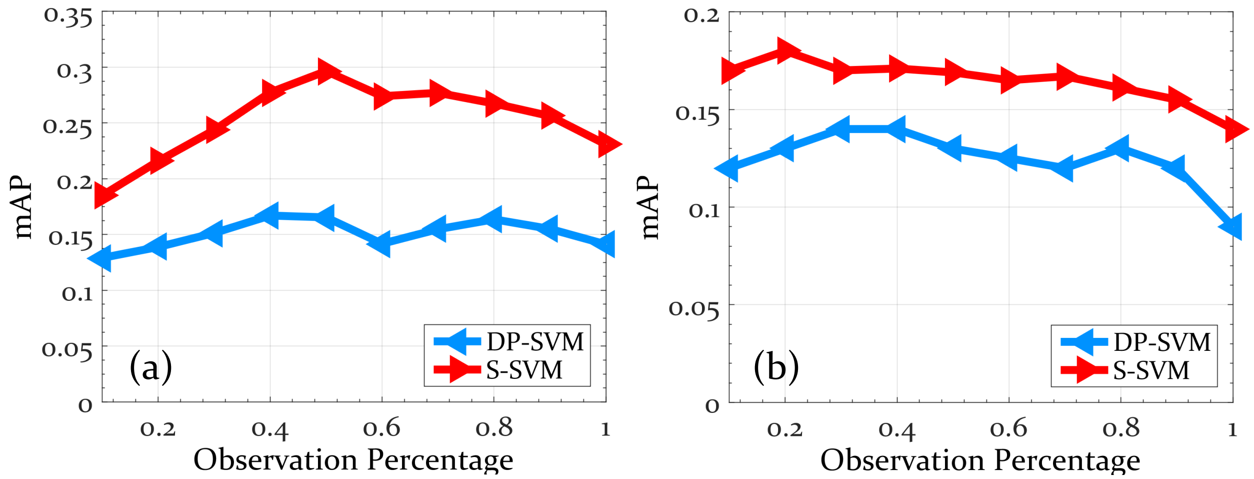


Figure 5.3: This figure shows online action (interaction) localization performance as a function of observed action percentage on (a) MSR-II and (b) UT-Interaction datasets. In contrast to Fig. 5.4(d,f), there are two important differences. First, mean average precision (mAP) is reported instead of multi-label prediction (classification). Furthermore, the evaluation is over untrimmed videos, and includes background clips.

The dataset consists of two sets, where each set has 10 video sequences and each sequence having at least one execution per interaction. Videos involve camera jitter with varying background, scale and illumination. We follow the recommended experimental setup by using 10-fold leave-one-out cross validation per set. That is, within each set we leave one sequence for testing and use remaining 9 for training. We report the average localization performance of the proposed approach using ROC @ 20% overlap and AUC curves.

5.3.2 Results and Analysis

Action (Interaction) Prediction with Time: The prediction accuracy (multi-label classification over ground truth action (interaction) tube) is evaluated with respect to the percentage of action (interaction) observed.

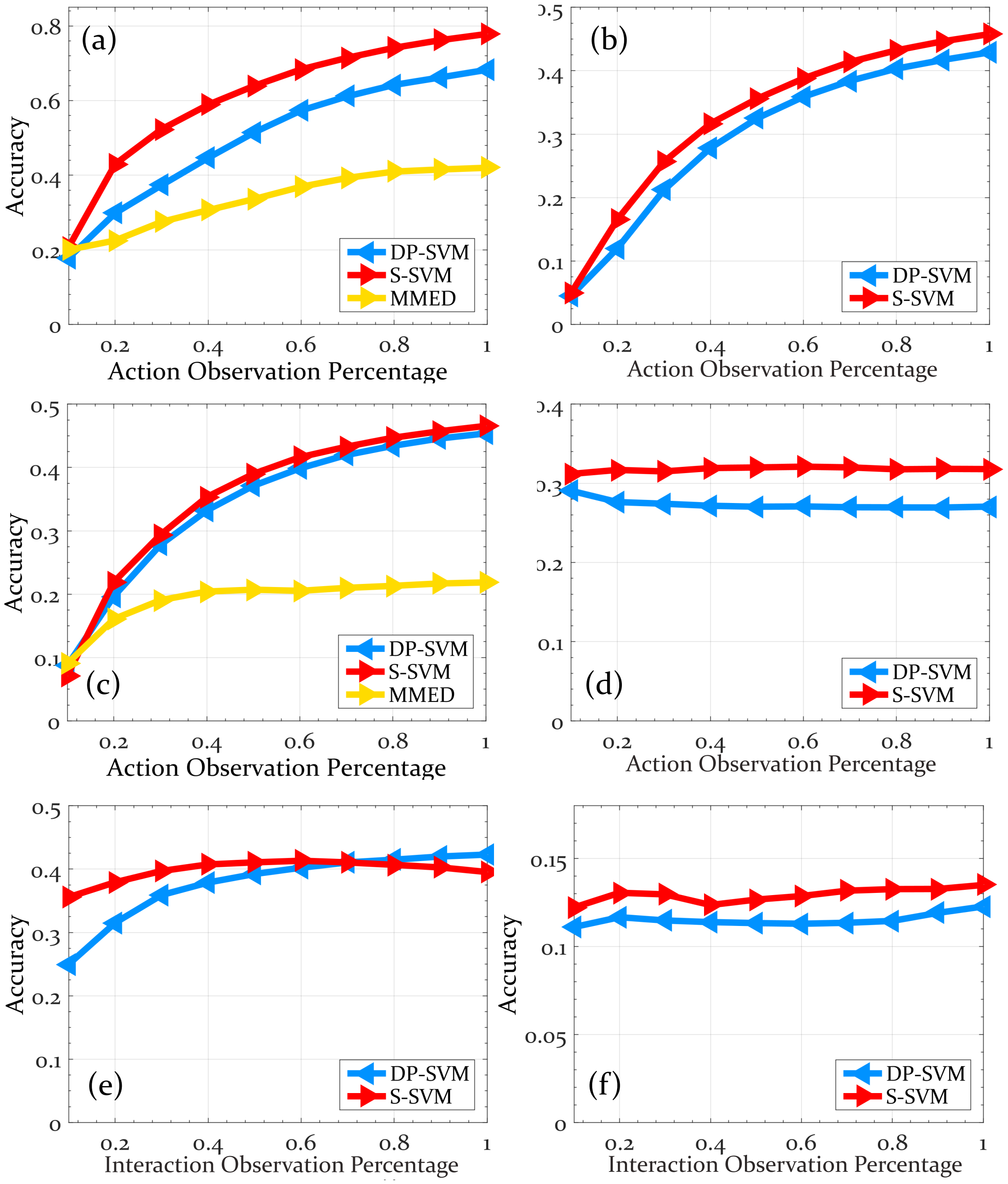


Figure 5.4: This figure shows action prediction accuracy as a function of observed percentage of action or interaction for (a) UCF Sports, (b) JHMDB, (c) sub-JHMDB, (d) MSR-II, (e) TV Human Interaction and (f) UT Interaction datasets. Prediction performance by the baseline Binary SVM with Dynamic Programming approach is shown in blue, and that of Structural SVM with the red curve. We compare the performance of our action prediction with MMED [25] (yellow curve) for UCF Sports and Sub-JHMDB datasets.

Fig. 5.4 shows the accuracy against time for (a) UCF Sports, (b) JHMDB, (c) sub-JHMDB, (d) MSR-II, (e) TV Human Interaction and (f) UT Interaction datasets, while Fig. 5.3 shows mean average precision evaluated on untrimmed videos of (a) MSR-II and (b) UT-Interaction datasets. The results show that Structural SVM in general performs better than Binary SVM with Dynamic Programming as it learns to predict higher confidence as more action is observed. It is evident that predicting the class of an action based on partial observation is very challenging, and the accuracy of correctly predicting the action increases as more information becomes available. However, the curves for MSR-II (Fig. 5.4(e)) and UT Interaction (Fig. 5.4(g)) datasets do not reflect noticeable change as more action (interaction) is observed. This is partially due to the reason that both these datasets have very few classes (3 and 6, respectively), and there is little confusion among classes from the onset of actions.

An analysis of prediction accuracy per action class is shown in Fig. 5.5 for (a) sub-JHMDB and (b) TV Human Interaction datasets. Similarly Fig. 5.6(a) shows per-action results for UCF Sports. A common theme among the results of all the datasets is that actions which have actors in upright standing position are always easy to predict and localize compared to other actions. This is also visible from the curves of *kicking* (UCF Sports), *kick ball* (sub-JHMDB) and *hand shake, high five* (TV Human Interaction) which begin with a high prediction accuracy and drop slightly as observation time period progresses, thus suggesting strong bias of classifier towards such actions (interactions). For sub-JHMDB, high prediction accuracy actions include *push* and *pull up*, both of which have humans in upright position making pose estimation easy, whereas *jump* is the most difficult action to predict. An inspection of videos for this action reveals that most of the instances were taken from parkour exhibiting high articulation and intra-class variation. For TV Human Interaction dataset, *hug* is easy to predict whereas *kiss* is the most difficult due to its subtle motion and high confusion with *hug*. For UCF Sports, high performing actions are *kicking, walking* and *running*, all upright with smooth motion of legs. For this dataset, the most difficult action is *swing* *side* due to high articulation with most of the instances depicting swinging of the sportsperson from the very first frame with different pose at the beginning of each action instance.

Finally, we also analyze the performance of DP-SVM and S-SVM in Fig. 5.6(b) which shows the difference of prediction accuracy DP-SVM and S-SVM. Longer duration actions such as *diving, walking, running, riding horse* gain significant boost in prediction accuracy, with average performance increasing by about 12% over the baseline DP-SVM for this dataset.

Since each action has its own predictability, we also analyze how early we can predict each action. We arbitrarily set the prediction accuracy to 30% and show the percentage of action observation required for each action of JHMDB, sub-JHMDB and UCF Sports datasets in Table 5.1. Although we set a reasonable prediction target, certain actions do not reach such prediction accuracy even until the completion of the video. This highlights the challenging nature of online action prediction and localization.

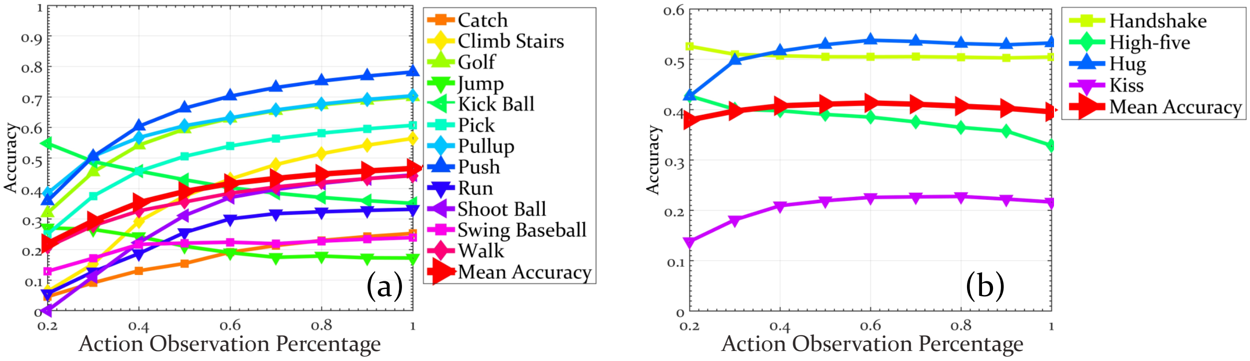


Figure 5.5: This figure shows per-action prediction accuracy as a function of observed action (interaction) percentage for (a) sub-JHMDB and (b) TV Human Interaction datasets. The mean accuracy for all actions (interactions) is shown with bold red curve.

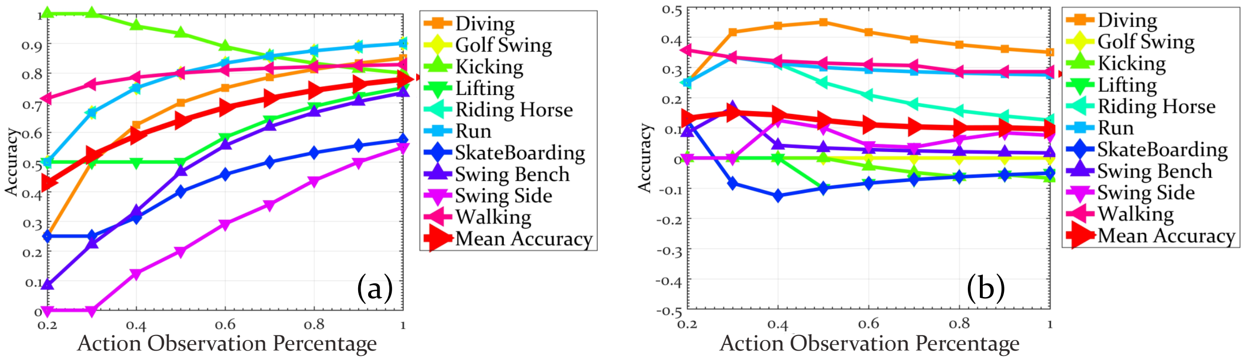


Figure 5.6: This figure shows per-action prediction accuracy as a function of observed action percentage for UCF Sports dataset for (a) Structural SVM approach (Sec. 5.2.2) and (b) and its difference with SVM and Dynamic Programming (Sec. 5.2.1). On average, S-SVM outperforms DP-SVM.

Action (Interaction) Localization with Time: To evaluate online performance, we analyze how the localization performance varies across time by computing prediction accuracy as a function of observed action (interaction) percentage. Fig. 5.7 shows the AUC against time for different overlap thresholds ($10\% - 60\%$) for (a) JHMDB, (b) sub-JHMDB, (c) MSR-II and (d) UCF Sports action datasets. The AUC as a percentage of observed interaction percentage is shown for (e) TV Human Interaction and (f) UT Interaction datasets as well. We compute the AUC with time in a cumulative manner such that the accuracy at 50% means localizing an action from start till one-half of the video has been observed. This gives an insight into how the overall localization performance varies as a function of time or observed percentage in testing videos. These graphs show that it is challenging to localize an action at the beginning of the video, since there is not enough discriminative motion observed by the algorithm to distinguish different actions. Furthermore, our approach first learns an appearance model from pose bounding boxes, which are improved and refined as time progresses. This improves the superpixel-based appearance confidence, which then improves the localization, and stabilizes the AUC. The curves also show that the AUC is inversely proportional to the overlap threshold.

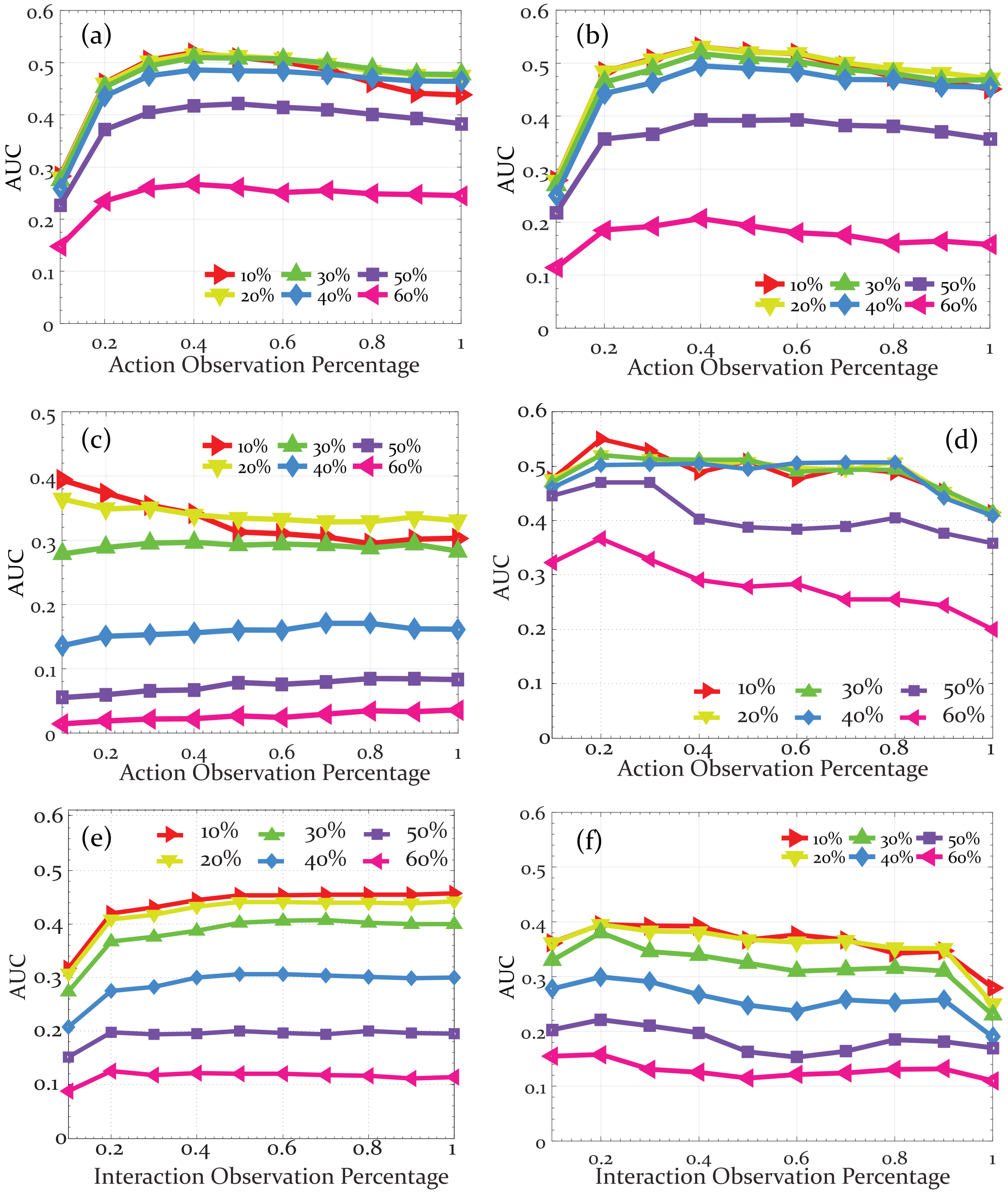


Figure 5.7: This figure shows online action (interaction) localization performance as a function of observed action percentage on (a) JHMDB, (b) sub-JHMDB, (c) MSR-II, (d) UCF-Sports, and as a function of observed interaction percentage for (e) TV Human Interaction and (f) UT Interaction datasets. Different curves show evaluations at different overlap thresholds: 10% (red), 30% (green) and 60% (pink).

Table 5.1: This table shows the the percentage of video observation required to achieve a prediction accuracy of 30%. Results in the first two rows are from JHMDB, then from sub-JHMDB and the last row is from UCF Sports dataset. Actions with missing values indicate that they did not reach a prediction accuracy of 30% until video completion.

JHMDB Actions	*Shoot Ball*	*Shoot Gun*	*Pull up*	*Golf*	*Clap*	*Climb Stairs*	*Shoot Bow*	*Brush Hair*	*Pour*	*Push*	*Walk*	
Video (%)	*1%*	*1%*	*16%*	*19%*	*25%*	*26%*	*28%*	*32%*	*32%*	*36%*	*36%*	
JHMDB Actions	*Sit*	*Swing Baseball*	*Run*	*Stand*	*Catch*	*Jump*	*Pick*	*Kick Ball*	*Throw*	*Wave*		
Video (%)	*40%*	*40%*	*48%*	*60%*	-	-	-	-	-	-		
sub-JHMDB Actions	*Kick Ball*	*Pullup*	*Golf*	*Push*	*Walk*	*Pick*	*Climb Stairs*	*Shoot Ball*	*Run*	*Catch*	*Jump*	*Swing Baseball*
Video (%)	*1%*	*17%*	*18%*	*18%*	*20%*	*24%*	*41%*	*48%*	*60%*	-	-	-
UCF Sports Actions	*Kicking*	*Lifting*	*Walking*	*Golf Swing*	*Riding Horse*	*Run*	*Diving*	*Swing Bench*	*Skate Boarding*	*Swing Side*		
Video (%)	*1%*	*1%*	*1%*	*15%*	*15%*	*15%*	*22%*	*36%*	*37%*	*61%*		

There are two interesting observations that can be made from these graphs. First, for the JHMDB, sub-JHMDB and MSR-II datasets in Fig. 5.7(a,b,c), the results improve initially, but then deteriorate in the middle, i.e. when the observation percentage is around 60%. The reason is that most of the articulation and motion happens in the middle of the video. Thus, the segments in the middle are the most difficult to localize, resulting in drop of performance. Second, the curves for UCF Sports in Fig. 5.7(d) depict a rather unexpected behavior in the beginning, where localization improves and then suddenly worsens at around 15% observation percentage. On closer inspection, we found that this is due to rapid motion in some of the actions, such as *diving* and *swinging (side view)*. For these actions, the initial localization is correct when the actor is stationary, but both actions have very rapid motion in the beginning, which violates the continuity constraints applicable to many other actions. This results in a drop in performance, and since this effect accumulates as observation percentage increases, the online algorithm never attains the peak again for many overlap thresholds despite observing the entire action.

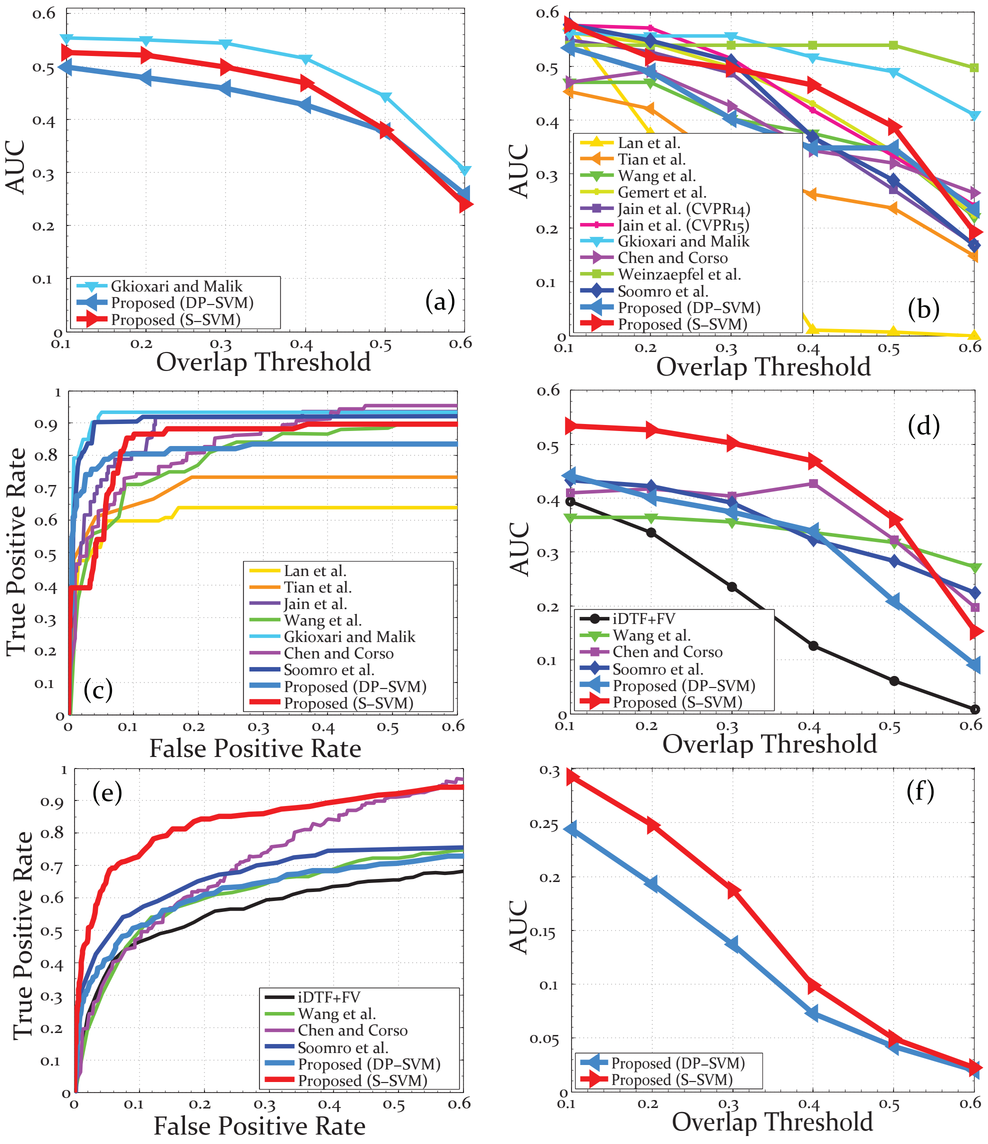


Figure 5.8: This figure shows action localization results of the baseline Binary SVM with Dynamic Programming (DP-SVM) and Structural SVM (S-SVM) approaches, along with existing *offline* methods on four action datasets (JHMDB, UCF Sports, sub-JHMDB and MSR-II). (a) shows AUC curves for JHMDB, while (b) and (c) show AUC and ROC @ 20%, respectively, for UCF Sports dataset. AUC and ROC @ 20% overlap are shown in (d) and (e) for sub-JHMDB dataset, finally AUC for MSR-II dataset is shown in (f). The curve for S-SVM method is shown in red and DP-SVM is shown in blue, while other *offline* localization methods including Lan *et al.* [46], Tian *et al.* [86], Wang *et al.* [95], van Gemert *et al.* [92], Jain *et al.* [30] [31], Gkioxari and Malik [20], Chen and Corso [11], Weinzaepfel *et al.* [102] and Soomro *et al.* [81] are shown with different colors. Despite being online, the proposed approach performs competitively overall compared to existing offline methods.

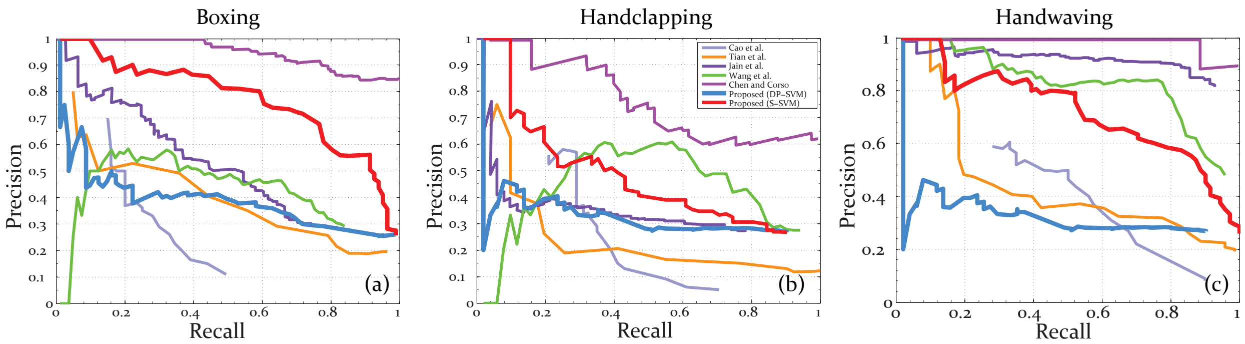


Figure 5.9: This figure shows action localization results on MSR-II dataset. The precision/recall curves are drawn for three actions: (a) boxing, (b) Hand clapping and (c) hand waving. We perform competitively to many existing *offline* methods. Red curve shows the proposed S-SVM approach, while blue curve shows the results of the baseline DP-SVM method.

Comparison with Offline Methods: We also evaluate the performance of our method against existing *offline* state-of-the-art action localization methods. Fig. 5.8(a) shows the results of the proposed S-SVM method, on JHMDB dataset, in red and the baseline DP-SVM in blue, while that of [20] in cyan. The difference in performance is attributed to the online vs. offline nature of the methods, as well as the use of CNN features by [20]. Quantitative comparison on UCF Sports using AUC and ROC @ 20% is shown in Fig. 5.8(b) and (c) respectively. Fig. 5.8 also shows the results of S-SVM and DP-SVM over all datasets where S-SVM outperforms DP-SVP highlighting the importance of structured prediction. The biggest gain in performance is visible in sub-JHMDB dataset, as shown by the AUC and ROC curves in Fig. 5.8(d) and (e), where despite being online S-SVM outperforms existing state-of-the-art methods.

For MSR-II dataset, we evaluate action localization and prediction using 1) precision/recall curve to draw comparison with existing methods as shown in Fig 5.9 for the three different actions: (a) boxing, (b) hand clapping and (c) hand waving, as well as through 2) AUC performance in Fig. 5.8 (f) for consistent evaluation with other datasets. The average precision per action is given in Table 5.2.

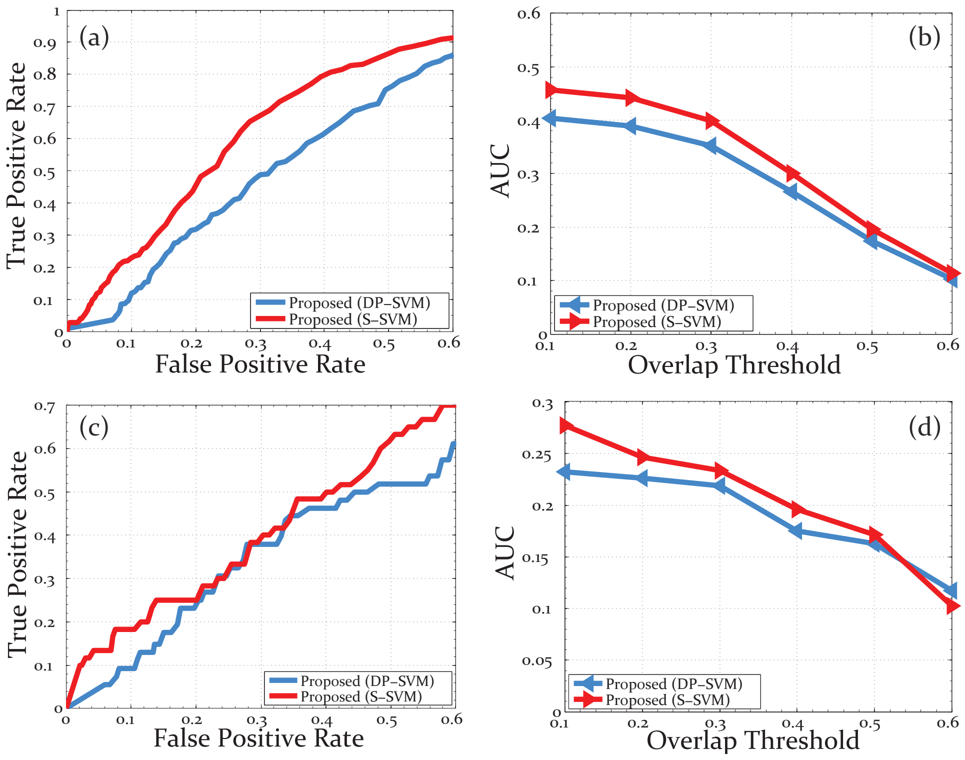


Figure 5.10: This figure shows interaction localization results on two interaction datasets. ROC @ 20% overlap and AUC curves for TV Human Interaction dataset are shown in (a) and (b), and for UT Interaction dataset in (c) and (d). In this figure, S-SVM shown in red and DP-SVM (baseline) in blue.

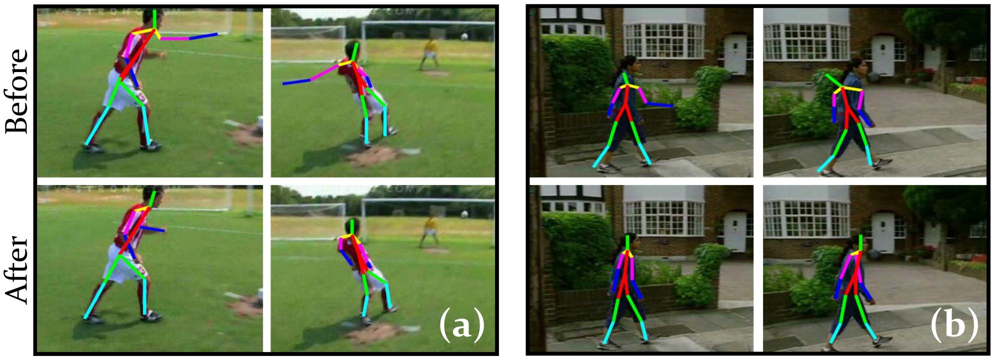


Figure 5.11: This figure shows qualitative results for pose refinement. Results show a comparison of raw poses (top row) and refined poses (bottom row) for (a) Kicking and (b) Walking.

Table 5.2: This table shows the average precision for MSR-II dataset on three different actions: (a) Boxing, (b) Handclapping and (c) Handwaving.

Method	*Boxing*	*Handclapping*	*Handwaving*
Cao et al. *[9]*	*17.5*	*13.2*	*26.7*
Tian et al. *[86]*	*38.9*	*23.9*	*44.4*
Jain et al. *[30]*	*46.0*	*31.4*	*85.8*
Wang et al. *[95]*	*41.7*	*50.2*	*80.9*
Chen and Corso [11]	*94.4*	*73.0*	*87.7*
Proposed (DP-SVM)	*37.3*	*28.3*	*42.9*
Proposed (S-SVM)	*75.3*	*43.4*	*71.3*

Table 5.3: This table shows the video mean average precision (mAP) for UCF Sports, JHMDB, Sub-JHMDB, TV Human Interaction and UT Interaction datasets.

Method	*UCF Sports*	*JHMDB*	*Sub JHMDB*	*TVHI*	*UT Interaction*
Saha et al. *[76]*	-	*71.5*	-	-	-
Peng and Schmid [65]	*94.8*	*73.1*	-	-	-
Weinzaepfel et al. *[102]*	*90.5*	*60.7*	-	-	-
Gkioxari and Malik [20]	*75.8*	*53.3*	-	-	-
DP-SVM	*65.4*	*50.2*	*43.6*	*38.5*	*17.0*
S-SVM	*70.3*	*51.9*	*49.3*	*42.8*	*20.3*

Generally, interaction datasets have either been used for classification [74], activity prediction [73] or video retrieval [62]. We are the first to evaluate online localization on these datasets. To keep evaluation metrics uniform, we present our performance on localization and prediction of human interactions in Fig. 5.10 using ROC and AUC curves for TV Human interaction (a,b) and UT interaction (c,d). We also report mean Average Precision (mAP) for all datasets in Table 5.3.

Pose Refinement: Pose-based foreground likelihood refines poses in an iterative manner using spatio-temporal smoothness constraints. Our qualitative results in Fig. 5.11 show the improvement in pose joint locations on two example videos..

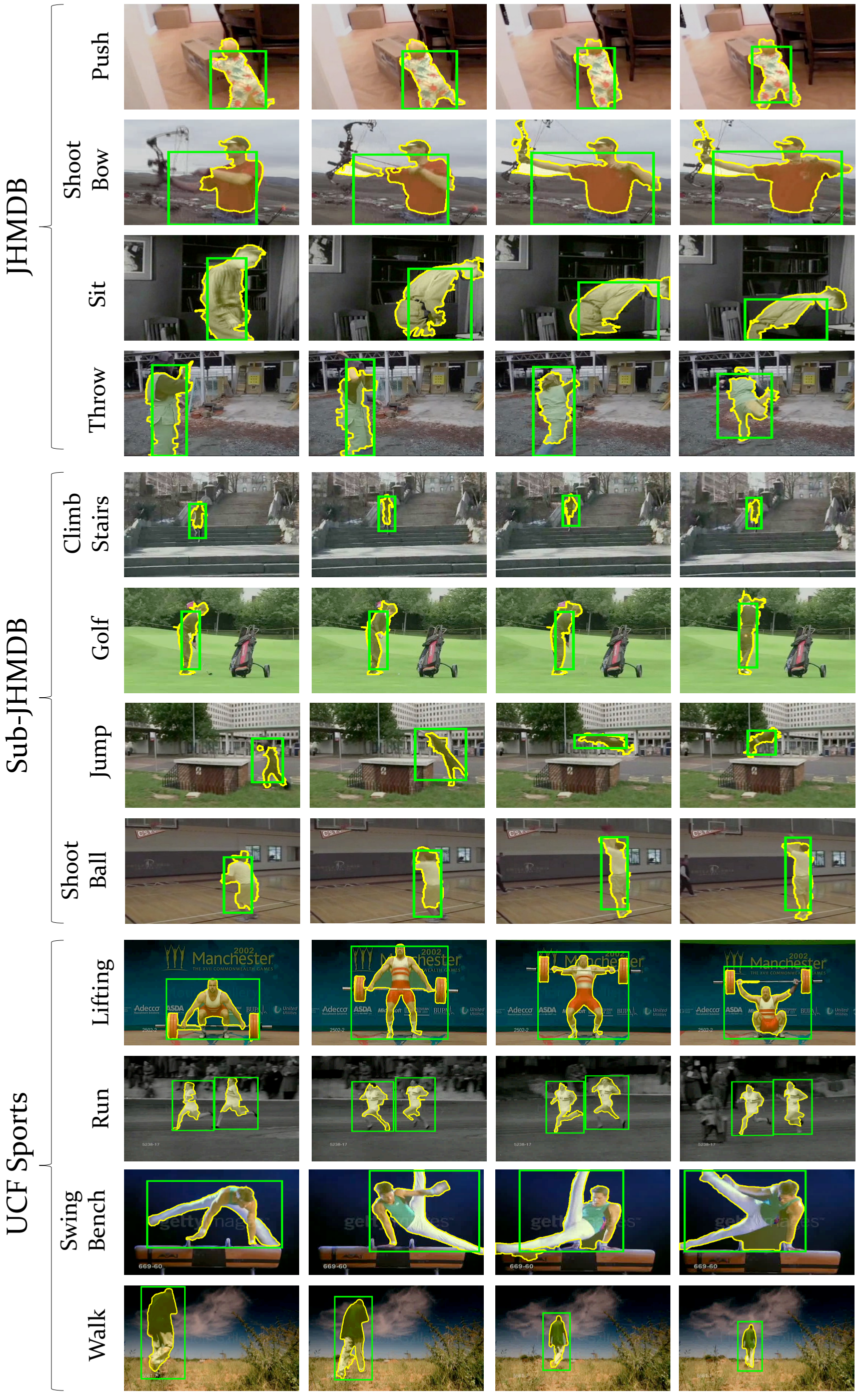


Figure 5.12: This figure shows qualitative results of the proposed approach for JHMDB, sub-JHMDB and UCF Sports datasets, where each action segment is shown with yellow contour and ground truth with green bounding box.

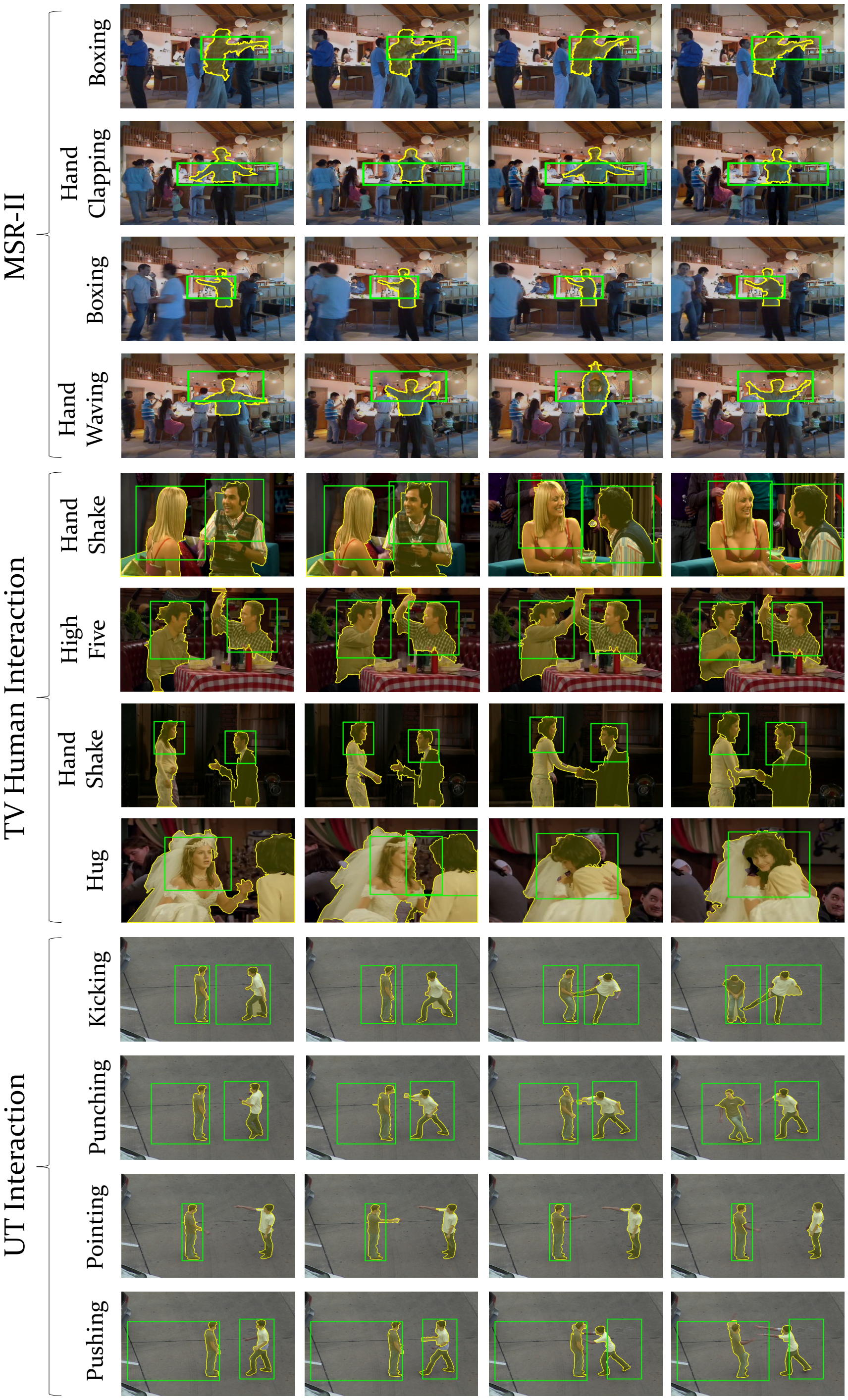


Figure 5.13: This figure shows qualitative results of the proposed approach for MSR-II, TV Human Interaction and UT Interaction datasets, where each action segment is shown with yellow contour and ground truth with green bounding box.

Action Segments: Since we use superpixel segmentation to represent the foreground actor, our approach outputs action segments. Our qualitative results in Fig. 5.12, 5.13 show the fine contour of each actor (yellow) along with the ground truth (green). Using superpixels and CRF, we are able to capture the shape deformation of the actors.

5.4 Summary

In this chapter, we introduced a new prediction problem of online action and interaction localization, where the goal is to simultaneously localize and predict action (interaction) in an online manner. We presented an approach which uses representations at different granularities - from high-level poses for initialization, mid-level features for generating action tubes, and low-level features such as iDTF for action (interaction) prediction. We also refine pose estimation in an online manner using spatio-temporal constraints. The localized tubes are obtained using CRF, and prediction confidences come from the classifier. We showed that the Structural SVM (S-SVM) formulation outperforms the dynamic programming with SVM (DP-SVM) hybrid. The intermediate results and analysis indicate that such an approach is capable of addressing this difficult problem, and performing competitive to some of the recent offline action localization methods. In the next chapter, we tackle the problem of action localization in an *unsupervised* manner. Supervised approaches require manual annotations of video level labels and frame-level bounding boxes, which can be quite time consuming and impractical with large number of videos. Hence, we propose to localize actions without any ground truth action class labels and annotations.

CHAPTER 6: UNSUPERVISED ACTION DISCOVERY AND LOCALIZATION

The approaches mentioned so far in the previous chapters (Chapter 4 and 5), address the problem of action localization in a *supervised* manner. They use manually labeled training videos, where bounding boxes are used to learn detectors, and action class labels are used to train classifiers, to localize and recognize an action, respectively. However, such efforts are time consuming and require hours of manual work to label the location (bounding box), class, and temporal boundaries of each action in a video. In addition, each action has its own complexity in terms of spatio-temporal deformations (i.e. in height, width, spatial location and temporal length). Also, the understanding of the temporal extent of an action is subjective, which may vary from person to person, and can lead to unwanted biases and errors. Given, such challenges and the abundance of unlabeled videos available on the Internet, *unsupervised* learning can provide a solution to the mentioned problems.

This chapter deals with two important topics: 1) *Unsupervised Action Discovery* and 2) *Unsupervised Action Localization.* We first present our discriminative clustering based action discovery approach. Then, we propose a novel Knapsack approach with spatio-temporal constraints, to select supervoxels in a video for action localization. This methodology involves joint action selectivity for training action classifiers and learns pairwise relations of supervoxels using Structural SVM. Next, we perform experimental evaluation on three challenging action datasets and show that our proposed *unsupervised* approach gives comparable performance to existing state-of-the-art *supervised* methods.

6.1 Action Discovery 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 Machine (SVM) classifer 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. 2).

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 [55], 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* [63, 64], 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 [87]), 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 $\mathbf{A}_k = a^{ij}_k$, where $a^{ij}_k = \omega_k(i, j)$ if $(i, j) \in \mathbf{E}_k$, and $a^{ij}_k = 0$ otherwise. As there are no self-loops in $\mathbf{G}_k$, the main diagonal of $\mathbf{A}_k$ is zero.

Algorithm 2 Algorithm to Discover K Action Classes

Input: Action Discovery Video Set $\mathcal{V}$

Output: Action Clusters $C_1 \ldots C_K$

```
procedure DISCOVER ACTION CLASSES($\mathcal{V}$)
    $C_1 \ldots C_K \Leftarrow spectral\_clustering(\mathcal{V})$    ▷ Cluster $V$ Videos using Ng et al. [55]
    $\Xi_1 \ldots \Xi_K \Leftarrow dominant\_sets(C_1 \ldots C_K)$    ▷ Find Dominant Sets for all $K$ Action Clusters
    $\mathbf{\Lambda} \Leftarrow \bigcup_{k=1}^{K} \widetilde{\Xi}_k$    ▷ Group Non-Dominant Sets from all $K$ Action Clusters
    $C_1 \ldots C_K \Leftarrow \Xi_1 \ldots \Xi_K$    ▷ Initialize all clusters to Dominant Sets
    do
        for $k = 1$ to $K$ do
            $\mathbf{\Omega}_k \Leftarrow svm\_train(C_k, \bigcup_{k'=1, k' \neq k}^{K} C_{k'})$    ▷ Train Classifier on each Action Cluster
            $C_k^{new} \Leftarrow select\_top(\mathbf{\Omega}_k, \mathbf{\Lambda}, \eta)$    ▷ Test on video set $\mathbf{\Lambda}$ to select top $\eta$ videos for each cluster
            $C_k \Leftarrow C_k \cup C_k^{new}$    ▷ Update all $K$ Clusters by adding newly classified videos from set $\mathbf{\Lambda}$
        end for
        $\mathbf{\Lambda} \Leftarrow \mathbf{\Lambda} \backslash \bigcup_{k=1}^{K} C_k^{new}$    ▷ Remove newly selected videos from test set $\mathbf{\Lambda}$
    while $\mathbf{\Lambda} \neq \emptyset$    ▷ Loop until all videos have been assigned to one of the clusters
    return $C_1 \ldots C_K$
end procedure
```

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:

$$\omega_{\Xi_k}(i) = \begin{cases} 1, & \text{if} |\Xi_k| = 1 \\ \sum_{j \in \Xi_k \setminus \{i\}} \phi_{\Xi_k \setminus \{i\}}(j,i) \omega_{\Xi_k \setminus \{i\}}(j), & \text{otherwise,} \end{cases} \tag{6.1}$$

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:

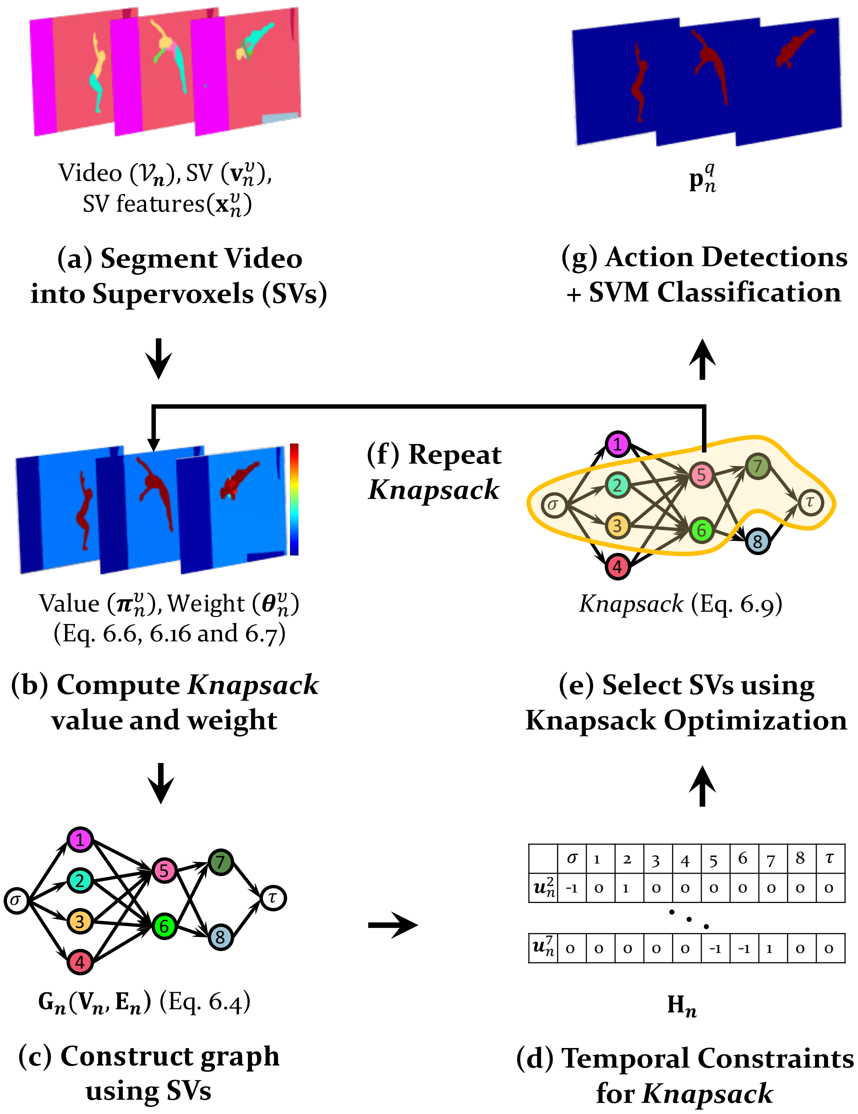
1. $\omega_{\Xi_k}(i) > 0$, for all $i \in \Xi_k$

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* [63, 64], arising from evolutionary game theory. As shown in Algorithm 2, 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.

6.2 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 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. 6.1).


Video ($\mathcal{V}_n$), SV ($\mathbf{v}_n^v$), SV features($\mathbf{x}_n^v$)
(a) Segment Video into Supervoxels (SVs)
$\mathbf{p}_n^q$
(g) Action Detections + SVM Classification
(f) Repeat *Knapsack*
1
2
3
4
5
6
7
8
σ
τ
Value ($\boldsymbol{\pi}_n^v$), Weight ($\boldsymbol{\theta}_n^v$) (Eq. 6.6, 6.16 and 6.7)
(b) Compute *Knapsack* value and weight
Knapsack (Eq. 6.9)
(e) Select SVs using Knapsack Optimization
$\mathbf{G}_n(\mathbf{V}_n, \mathbf{E}_n)$ (Eq. 6.4)
(c) Construct graph using SVs
$\mathbf{H}_n$
(d) Temporal Constraints for *Knapsack*

	σ	1	2	3	4	5	6	7	8	τ
$\boldsymbol{u}_n^2$	-1	0	1	0	0	0	0	0	0	0
					⋱					
$\boldsymbol{u}_n^7$	0	0	0	0	0	-1	-1	1	0	0

Figure 6.1: 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 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 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.

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 $\mathbf{x}_n^\upsilon = \{{}_1\mathbf{x}_n^\upsilon \ldots {}_R\mathbf{x}_n^\upsilon\}$, 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:

$$\mathbf{Z}_n(\upsilon, \upsilon') = \begin{cases} 1, & \text{if } \mathbf{v}_n^\upsilon \in \mathcal{N}_{\mathbf{G}_n}(\mathbf{v}_n^{\upsilon'}) \ \& \ f_{start}(\mathbf{v}_n^{\upsilon'}) > f_{start}(\mathbf{v}_n^\upsilon) \ \& \ f_{end}(\mathbf{v}_n^{\upsilon'}) > f_{end}(\mathbf{v}_n^\upsilon) \\ 0, & \text{otherwise,} \end{cases} \tag{6.2}$$

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* [23] and 3) *Motion Boundary* [94].

Given a video $\mathcal{V}_n$, we use Faster-RCNN [70] 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 $\mathbf{\Gamma}_n = \{\mathbf{\Gamma}_n^1 \ldots \mathbf{\Gamma}_n^{F_n}\}$, 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:

$$\mathbf{S}_{hm}(\mathbf{v}_n^{\upsilon}, \mathcal{B}_n, \boldsymbol{\Gamma}_n) = \rho^{-1} \sum_{f=1}^{F_n} \arg\max_{b} {}_b\boldsymbol{\Gamma}_n^f \left(\frac{{}_f\mathbf{v}_n^{\upsilon} \cap {}_b\mathcal{B}_n^f}{|\,{}_f\mathbf{v}_n^{\upsilon}|} \right), \tag{6.3}$$

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:

$$\boldsymbol{\Pi}(\mathbf{v}_n^{\upsilon}, \mathcal{B}_n, \boldsymbol{\Gamma}_n, \mathbf{x}_n^{\upsilon}) = \gamma_{hm}\mathbf{S}_{hm}(\mathbf{v}_n^{\upsilon}, \mathcal{B}_n, \boldsymbol{\Gamma}_n) + \gamma_{sal}\mathbf{S}_{sal}(\mathbf{v}_n^{\upsilon}, \mathbf{x}_n^{\upsilon}) + \gamma_{mb}\mathbf{S}_{mb}(\mathbf{v}_n^{\upsilon}, \mathbf{x}_n^{\upsilon}), \tag{6.4}$$

where $\mathbf{S}_{sal}(.)$ and $\mathbf{S}_{mb}(.)$ are the functions to compute supervoxel saliency and motion boundary, respectively. The associated weights in Eq. 6.4 are symbolized by γ. Finally, the supervoxel value is given by $\boldsymbol{\pi}_n^{\upsilon} = \boldsymbol{\Pi}(\mathbf{v}_n^{\upsilon}, \mathcal{B}_n, \boldsymbol{\Gamma}_n)$.

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

$$\boldsymbol{\Theta}_n = \mathcal{O}^{-1} \sum_{f=1}^{F_n} \sum_{b=1}^{\mathcal{O}} |\,{}_b\mathcal{B}_n^f|, \tag{6.5}$$

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. 6.2. Fig. 6.1(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 $\mathbf{v}_n^\upsilon \in \mathcal{V}_n$ a binary label variable $\mathbf{u}_n^\upsilon$, which is 1 if $\mathbf{v}_n^\upsilon$ 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:

$$\begin{gathered} \underset{\mathbf{u}_n}{\text{maximize}} \sum_{m=1}^{M+2} \boldsymbol{\pi}_n^m \mathbf{u}_n^m \text{ subject to } \sum_{m=1}^{M+2} \boldsymbol{\theta}_n^m \mathbf{u}_n^m \leq \boldsymbol{\Theta}, \\ \mathbf{u}_n^m \in \{0,1\}, \quad \mathbf{u}_n^\sigma = 1, \quad \mathbf{u}_n^\tau = 1, \quad \mathbf{H}_n \mathbf{u}_n \leq 0. \end{gathered} \tag{6.6}$$

This function optimizes over supervoxels to select the set of supervoxels having maximum value, while satisfying temporal order and the weight limit.

Action Annotations: Since, each *knapsack* solution gives an annotated action, we recursively generate multiple annotations $\mathbf{p}_n^q = \bigcup_{\upsilon=1}^M \mathbf{v}_n^\upsilon, \quad \forall \mathbf{u}_n^\upsilon \neq 0$, where $q = \{1 \dots Q_n\}$ 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.

6.2.1 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:

$$\begin{aligned} &\underset{\mathbf{r}_n}{\text{minimize}} && \mathbf{r}_n^T\mathbf{U}\mathbf{r}_n - \mathbf{P}\mathbf{r}_n, \\ &\text{subject to} && \mathbf{r}_n \in \{0,1\}, \forall \mathcal{V}_n \in \mathcal{V} : \sum_{q=1}^{Q_n} \mathbf{r}_n^q = 1, \end{aligned} \tag{6.7}$$

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. 6.7 is non-convex, we make it convex by taking the normalized laplacian [80] 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. We also 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.

6.3 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*.

6.3.1 Training Action Classifiers

Knapsack approach to action annotation selects supervoxels by maximizing the sum of individual scores to annotate actions. These scores in Eq. 6.4 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. 6.2.1) in the training data.

SVM for Unary Learning: We learn a discriminative SVM classifier by using supervoxels within selected action annotations (in Eq. 6.7) as positive examples and the rest as negative examples.

Structural SVM for Pairwise Learning: Let $\mathbf{v}_n^{v'} \in \mathcal{N}_{\mathbf{G}_n}(\mathbf{v}_n^v)$ belong to the neighborhood of $\mathbf{v}_n^v$. We gather such pairwise relations from training videos and their annotations (in Eq. 6.7) to propose a Structural Support Vector Machine (S-SVM) formulation with margin re-scaling construction, which captures the relations between foreground-background as well as within foreground action using structured labels, as follows:

$$\begin{aligned}
\underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_{l=1}^{N*M} \xi_l, \\
\text{subject to} \quad & \langle \mathbf{w}, \boldsymbol{\Psi}_l([\mathbf{x}_l \mathbf{x}_{l'}], \mathbf{y}_l)\rangle - \langle \mathbf{w}, \boldsymbol{\Psi}_l([\mathbf{x}_l \mathbf{x}_{l'}], \mathbf{y})\rangle \geq \boldsymbol{\Delta}(\mathbf{y}_l, \mathbf{y}) - \xi_l, \\
& \qquad \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_l, \xi_l \geq 0, \forall l,
\end{aligned} \tag{6.8}$$

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.6.8 are defined as:

$$\mathcal{Y} = \begin{cases} -1, & \mathbf{v}^l \notin \boldsymbol{\kappa} \wedge \mathbf{v}^{l'} \notin \boldsymbol{\kappa} \\ 0, & \mathbf{v}^l \in \boldsymbol{\kappa} \wedge \mathbf{v}^{l'} \notin \boldsymbol{\kappa} \\ 1, & \text{otherwise,} \end{cases} \tag{6.9}$$

where $\boldsymbol{\kappa} = \bigcup_{n=1}^{N} \bigcup_{q=1}^{Q} \mathbf{p}_n^q$. The loss function in Eq.6.8 is defined as:

$$\boldsymbol{\Delta}(\mathbf{y}_l, \mathbf{y}_{l'}) = \begin{cases} |\mathbf{y}_l - \mathbf{y}_{l'}|, & \mathbf{v}^l \notin \boldsymbol{\kappa} \wedge \mathbf{v}^{l'} \notin \boldsymbol{\kappa} \\ \zeta + \varepsilon, & \mathbf{v}^l \in \boldsymbol{\kappa} \wedge \mathbf{v}^{l'} \notin \boldsymbol{\kappa} \\ \varepsilon, & \text{otherwise,} \end{cases} \tag{6.10}$$

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 $\boldsymbol{\psi}_{\mathcal{P}} : \mathcal{X} \mapsto \mathcal{Y}$ that scores a pair of supervoxels in the testing video as:

$$\boldsymbol{\psi}_{\mathcal{P}}([\mathbf{x}_t \mathbf{x}_{t'}]) = \arg\max_{y \in \mathcal{Y}} \langle \mathbf{w}, \boldsymbol{\Psi}_t([\mathbf{x}_t \mathbf{x}_{t'}], \mathbf{y}_t) \rangle. \tag{6.11}$$

6.3.2 Testing using Knapsack Localization

In a testing video $\mathcal{V}_s$, we compute supervoxels $\mathcal{V}_s = \{\mathbf{v}_s^1 \dots \mathbf{v}_s^T\}$, where $t = \{1 \dots T\}$ 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. 6.2 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. 6.6. Since, we are able to learn the unary and pairwise relatons between supervoxels from action annotations in training videos, we use the following updated function to compute supervoxel action distinctness:

$$\boldsymbol{\Pi}(\mathbf{v}_s^t, \mathcal{B}_s, \boldsymbol{\Gamma}_s, \mathbf{x}_s^t) = \gamma_{hm}\mathbf{S}_{hm}(\mathbf{v}_s^t, \mathcal{B}_s, \boldsymbol{\Gamma}_s) + \gamma_{sal}\mathbf{S}_{sal}(\mathbf{v}_s^t, \mathbf{x}_s^t) + \gamma_{mb}\mathbf{S}_{mb}(\mathbf{v}_s^t, \mathbf{x}_s^t) \\ + \gamma_{\mathcal{U}}\boldsymbol{\Upsilon}_{\mathcal{U}}(\mathbf{v}_s^t, \mathbf{x}_s^t) + \gamma_{\mathcal{P}}\boldsymbol{\Upsilon}_{\mathcal{P}}(\mathbf{v}_s^t, \mathbf{x}_s^t, \mathbf{G}_s), \quad (6.12)$$

where $\Upsilon_{\mathcal{U}}(.)$ and $\Upsilon_{\mathcal{P}}(.)$ are the unary and pairwise functions, respectively. The weights in Eq. 6.12 are symbolized by γ. The pairwise function is an accumulation of neighboring relations $\boldsymbol{\Upsilon}_{\mathcal{P}}(\mathbf{v}_s^t, \mathbf{x}_s^t, \mathbf{G}_s) = \varrho^{-1}\sum_{t'=1}^{\mathcal{N}_{\mathbf{G}_n}(\mathbf{v}_s^t)} \boldsymbol{\psi}_{\mathcal{P}}([\mathbf{x}_t\mathbf{x}_{t'}])$, where ϱ is a normalizing constant.

6.4 Experimental Results and Analysis

We evaluate our *Unsupervised Action Discovery and Localization* approach on five challenging datasets: 1) UCF Sports [71, 83] 2) JHMDB [32], 3) Sub-JHMDB [32] 4) THUMOS13 [33], 5) UCF101 [84]. 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 [87] to cluster them into action classes with η=2. For action localization, we extract improved dense trajectory features (iDTFs) [94] for all videos. This is followed by supervoxel segmentation [59], which are encoded using Fisher [66] 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. 6.4 and 6.12 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 [46] 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.

Table 6.1: 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 [87] and iDTF [94] features on UCF Sports.

	UCF Sports		*Sub JHMDB*	*JHMDB*	*THUMOS 13*	*UCF101*
	iDTF	*C3D*				
K-Means	*31.5*	*51.1*	*39.3*	*37.2*	*46.7*	*45.4*
K-Medoids	*26.4*	*57.8*	*36.6*	*34.2*	*52.1*	*33.0*
S&M [80]	*40.5*	*57.2*	*38.7*	*37.9*	*54.4*	*6.5*
DS [64]	*25.9*	*46.3*	*24.5*	*12.1*	*50.1*	*5.6*
SC [55]	*53.1*	*69.6*	*46.1*	*45.6*	*77.2*	*51.6*
DAKM [37]	*58.4*	*73.6*	*47.6*	*45.2*	*82.5*	*37.1*
Proposed	***65.7***	***90.1***	***57.4***	***53.7***	***88.3***	***61.2***

Unsupervised Action Discovery: The proposed action discovery approach is tested on the training videos of five datasets. The results for clustering accuracy using the evaluation metrics used in [37] are reported in Table 6.1. Clustering on all datasets has been performed using C3D features, except for UCF Sports where we also report results using iDTF features for comparison. The number of clusters for each dataset are set to the number of action classes. We compare the performance of our approach with: K-Means, K-Medoids, Shi and Malik (S&M) [80], Dominant Sets (DS) [64], Spectral Clustering (SC) [55] and the state-of-the-art DAKM [37] clustering methods. As can be seen from the table, 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 [80] don't perform well. Significant improvement over *Dominant Sets* [64] and Spectral Clustering [55] 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.

Table 6.2: This table shows comparison of localization performance with weakly-supervised approach [52] on UCF Sports.

Actions	Dive	Golf	Kick	Lift	Ride	Run	Skate	Swing-B	Swing-S	Walk	**Average**
Ma et al. *[52]*	*44.3%*	*50.5%*	***48.3%***	*51.4%*	***30.6%***	*33.1%*	*38.5%*	***54.3%***	*20.6%*	*39.0%*	*41.0%*
Proposed (Weakly)	***59.4%***	***59.9%***	*37.7%*	***59.5%***	*14.1%*	***50.0%***	***57.9%***	*50.0%*	***44.6%***	***43.4%***	***47.7%***

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 our *Knapsack* annotation approach by performing significantly better ($\sim 7\%$) than published state-of-the-art weakly-supervised method of Ma *et al.* [52] in Table 6.2.

Unsupervised Action Localization: We show localization performance using AUC curves for (a) UCF Sports (b) JHMDB, (c) sub-JHMDB and (d) THUMOS13 in Fig. 6.2. 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 [46, 86] on UCF Sports dataset. *Supervised* baseline results have been reported by Wang *et al.* [95] on Sub-JHMDB and Soomro *et al.* [81, 82] on UCF Sports, JHMDB and THUMOS13 datasets. 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. 6.4, 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 as shown in Fig. 6.4.

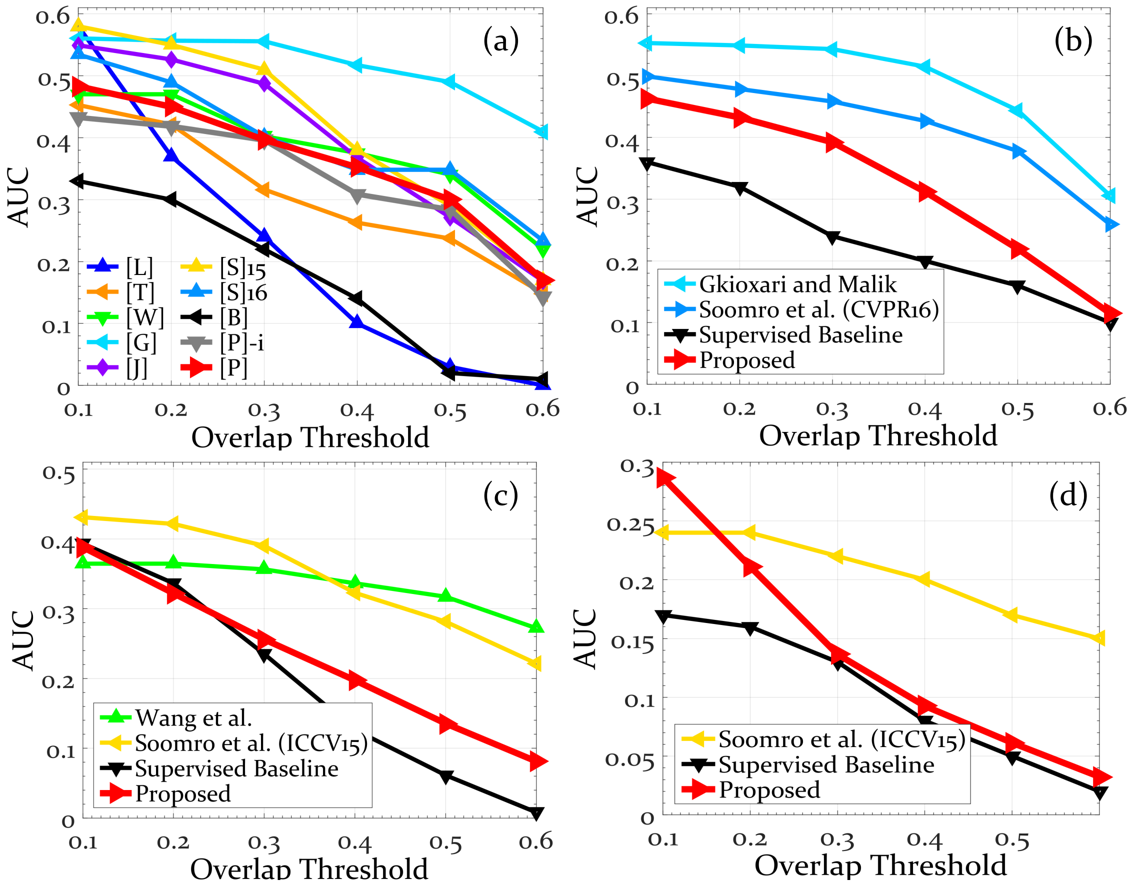


Figure 6.2: This figure shows AUC of the proposed *Unsupervised Action Localization* approach, along with existing *supervised* methods on (a) UCF Sports, (b) JHMDB, (c) Sub-JHMDB and (d) THUMOS13. The curves for the [P]roposed method is shown in red and *supervised* [B]aseline in black, while other *supervised* localization methods including [L]an *et al.* [46], [T]ian *et al.* [86], [W]ang *et al.* [95], [G]kioxari and Malik [20], [J]ain *et al.* [30], [S]oomro *et al.* [81, 82] 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 [94] features.

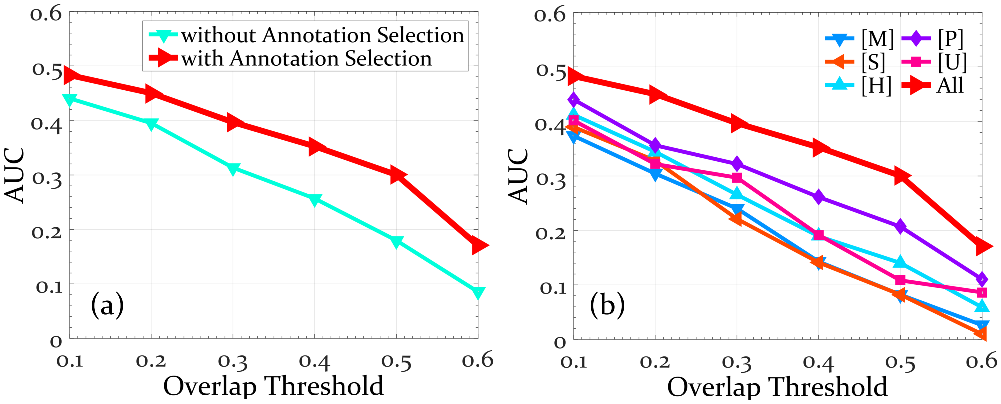


Figure 6.3: This figure shows the contribution of (a) Joint Annotation Selection (Eq. 6.7) and (b) individual components in computing action distinctness score for *Knapsack* value (Eq. 6.12) using AUC on UCF Sports. It includes [M]otion Boundary, [S]aliency, [H]umanness, [P]airwise S-SVM, [U]nary SVM and a combination of All i.e. M+S+H+U+P.

Feature Comparison: We show a comparison of the proposed *action discovery* approach using C3D and iDTF features in Table 6.1. 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 [87], 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. 6.2 (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. 6.3, 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. 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. 6.3(a). Fig. 6.3(b) shows the contribution of each component in action distinctness score (Eq. 6.12), 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.2 (∼2min), *Knapsack* in Eq.6.6 (∼1min) and Joint Annotation Selection in Eq.6.7 (∼1.1min), using an unoptimized MATLAB code running on an Intel Xeon E5645@2.4 Ghz/40GB RAM.

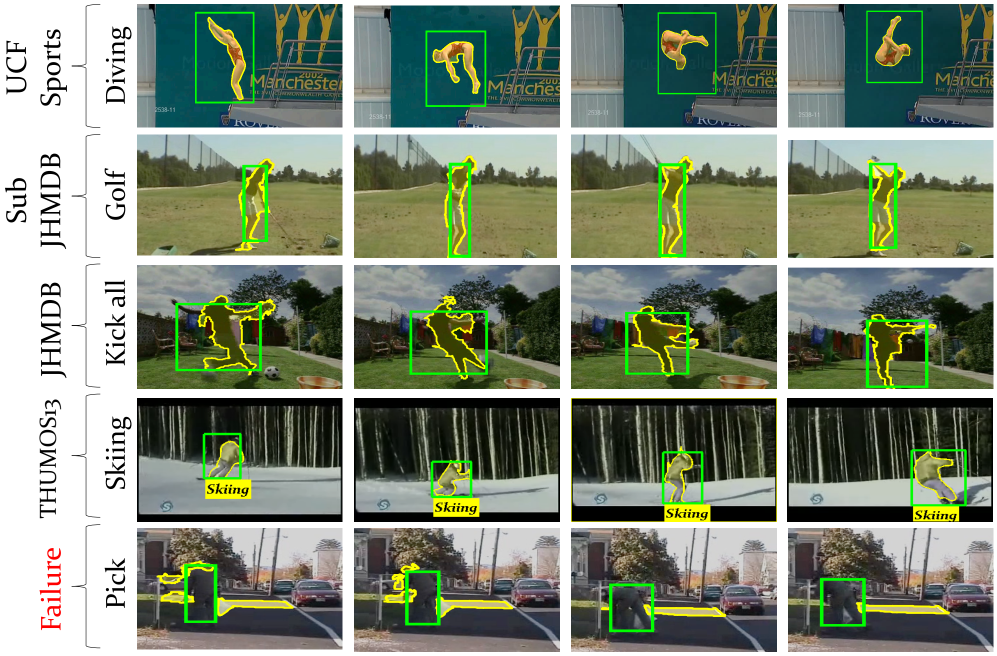


Figure 6.4: This figure shows qualitative results for the proposed approach on UCF Sports, Sub-JHMDB, JHMDB and THUMOS13 datasets (top four rows). Last row shows failure case from JHMDB dataset. The action localization is shown by yellow contour and ground truth bounding box in green.

6.5 Summary

In this chapter, we automatically discovered video action class labels and annotations to address the new problem of *Unsupervised Action Localization*. The presented approach discovers action classes, by using a discriminative clustering approach, where it iteratively selects videos to improve clustering. Actions are annotated using novel *knapsack* optimization for supervoxel selection. This optimization uses volume constraints to enforce the combined selection of supervoxels to be consistent with human spatial size and temporal extent of the action. Additionally, temporal constraints in this optimization ensure that the action is contiguous and spatio-temporally well-connected. Lastly, in testing videos we use a similar *knapsack* approach to detect actions and recognize them using SVM classifier learnt from our discovered labels and annotated actions.

CHAPTER 7: CONCLUSION AND FUTURE WORK

This dissertation addresses the problem of action localization in videos. Action detection methods rely on sliding window approach to localize actions, which is time consuming and impractical. However, the use of context information has helped reduce search space, as highlighted in Chapter 4. Timely, prediction and localization of actions (interactions) is shown in Chapter 5, using a person-centric foreground likelihood approach. Lastly, to save the manual effort of annotating videos, an unsupervised action localization approach is described in Chapter 6.

An efficient and effective approach to localize actions in videos is presented in Chapter 4. The approach starts with a random supervoxel and finds similar supervoxels from the training data to transfer the relative spatio-temporal location of an action to the video. Generating a conditional distribution over the sueprvoxel graph in the testing video, the next supervoxel is selected with the highest probability. This Context Walk is repeated for several steps, increasing the probability of predicting the action at each step. CRF is used to infer the location of the action and SVM is used to evaluate the action proposal. The use of supervoxels and context helps evaluate the classifier at fewer locations.

A new problem of predicting and localizing actions (interactions) is introduced in Chapter 5. A multi-level action representation is used in this approach, which includes: high-level poses, mid-level features for action tubes, and low-level iDTF features for prediction. Pose estimation is refined in an online manner using spatio-temporal constraints. CRF is used to localize action tubes and action is predicted using a Structural-SVM and dynamic programming with SVM approach. The approach performs competitive to the existing *offline* state-of-the-art methods.

Supervised approaches require tedious manual annotations of video action labels and bounding boxes. Therefore, we solve the problem of *unsupervised action localization* in Chapter 6. The presented approach discovers action classes, using a discriminative clustering approach, and actions are localized, using a modified *knapsack* approach to select supervoxels using spatio-temporal constraints.

We also introduce a challenging UCF101 dataset for action recognition and localization. The dataset collected from YouTube, consists of 101 action classes, with over 13k video clips. The videos are unconstrained with challenges such as poor lighting, cluttered background, severe camera motion and multiple instances of an action.

For future research, we highlight some extensions and areas of research that can improve the performance of the proposed approaches:

The supervoxels in Chapter 4 and the superpixels in Chapter 5 are evaluated independently to assign an actionness score, in a testing video. However, the superpixel (or supervoxel) merging criteria can be learnt from training data. In a testing video, a spatio-temporal graph can be used to determine combinatorially, which set of superpixels (or supervoxels) should be grouped together. This can further reduce the search space as only selected superpixels (supervoxels) would fit the merging criteria.

The online localization of actions (interactions) predicts the label at the current frame. However it doesn't predict how the action would unfold in future frames. It would be interesting to use a generative model that can reconstruct future frames based on the training data, as well as the video streamed until the current time step.

The *unsupervised action localization* approach in Chapter 6 uses an unlabeled, un-annotated set of videos to discover and localize actions. However, the underlying assumption is that the total number of possible actions is known and each video has to belong to one of these action classes. It would be worthwhile to explore the problem of unsupervised action discovery and localization from noisy videos, where the goal is to robustly cluster actions by removing any outlier action classes. This would provide a more realistic setting and has application towards anomaly or abnormal action class detection.

LIST OF REFERENCES

[1] K-lite codec package. http://codecguide.com/.

[2] Trecvid multimedia event recounting evaluation track. `http://www.nist.gov/itl/iad/mig/mer.cfm`.

[3] Youtube. http://www.youtube.com/.

[4] 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.

[5] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI*, 34(11), 2012.

[6] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3), 2011.

[7] B. Alexe, N. Hees, Y. Teh, and V. Ferrari. Searching for objects driven by context. In *NIPS*, 2012.

[8] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes, 2005. ICCV.

[9] L. Cao, Z. Liu, and T. S. Huang. Cross-dataset action detection. In *CVPR*, 2010.

[10] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. M. Siskind, and S. Wang. Recognize human activities from partially observed videos. In *CVPR*, 2013.

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

[12] W. Chen, C. Xiong, R. Xu, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *CVPR*, 2014.

[13] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition.

In *ICCV*, 2015.

[14] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010.

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

[16] A. Dehghan, H. Idrees, and M. Shah. Improving semantic concept detection through the dictionary of visually-distinct elements. In *CVPR*, 2014.

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

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

[19] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 32(9), 2010.

[20] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015.

[21] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, 2007.

[22] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009.

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

[24] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*. 2008.

[25] M. Hoai and F. De la Torre. Max-margin early event detectors. *IJCV*, 107(2), 2014.

[26] M. Hoai and A. Zisserman. Talking heads: Detecting humans and recognizing their interactions. In *CVPR*, 2014.

[27] 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.

[28] D.-A. Huang and K. M. Kitani. Action-reaction: Forecasting the dynamics of human inter-

action. In *ECCV*. Springer, 2014.

[29] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *ECCV*. 2010.

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

[31] 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.

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

[33] Y. 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.

[34] 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.

[35] G. Johansson, S. Bergstrom, and W. Epstein. Perceiving events and objects, 1994. Lawrence Erlbaum Associates.

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

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

[38] 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.

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

[40] Y. Kong and Y. Fu. Modeling supporting regions for close human interaction recognition. In *ECCV*, 2014.

[41] Y. Kong, Y. Jia, and Y. Fu. Learning human interaction by interactive phrases. In *ECCV*,

2012.

[42] Y. Kong, Y. Jia, and Y. Fu. Interactive phrases: Semantic descriptions for human interaction recognition. *IEEE TPAMI*, 36(9), 2014.

[43] Y. Kong, D. Kit, and Y. Fu. A discriminative model with multiple temporal scales for action prediction. In *ECCV*. 2014.

[44] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[45] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *ECCV*. 2014.

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

[47] K. Li and Y. Fu. Prediction of human activity by discovering temporal sequence patterns. *IEEE TPAMI*, 36(8), 2014.

[48] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu. Online human action detection using joint classification-regression recurrent neural networks. In *ECCV*, 2016.

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

[50] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild, 2009. CVPR.

[51] J. Lu, J. J. Corso, et al. Human action segmentation with hierarchical supervoxel consistency. In *CVPR*, 2015.

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

[53] S. Majiwa, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR*, 2011.

[54] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[55] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *NIPS*, 2, 2002.

[56] J. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classication, 2010. ECCV.

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

[58] M. Norouzi, A. Punjani, and D. J. Fleet. Fast search in hamming space with multi-index hashing. In *CVPR*, 2012.

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

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

[61] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman. Structured learning of human interactions in tv shows. *IEEE TPAMI*, 34(12), 2012.

[62] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. D. Reid. High five: Recognising human interactions in tv shows. In *BMVC*, 2010.

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

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

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

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

[67] H. Pirsiavash, C. Vondrick, and A. Torralba. Assessing the quality of actions. In *ECCV*. 2014.

[68] M. Raptis and L. Sigal. Poselet key-framing: A model for human activity recognition. In

CVPR, 2013.

[69] K. Reddy and M. Shah. Recognizing 50 human action categories of web videos, 2012. Machine Vision and Applications Journal (MVAP).

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

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

[72] P. Rota, N. Conci, N. Sebe, and J. M. Rehg. Real-life violent social interaction detection. In *ICIP*, 2015.

[73] M. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, 2011.

[74] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009.

[75] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA). http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html, 2010.

[76] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *BMVC*, 2016.

[77] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei. Spatial-temporal correlatons for unsupervised action classification. In *Workshop on Motion and video Computing*, 2008.

[78] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004.

[79] 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.

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

[81] K. Soomro, H. Idrees, and M. Shah. Action localization in videos through context walk. In

ICCV, 2015.

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

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

[84] 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.

[85] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009.

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

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

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

[89] D. Tran, J. Yuan, and D. Forsyth. Video event detection: From subvolume localization to spatiotemporal path search. *IEEE TPAMI*, 36(2), 2014.

[90] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2), 2013.

[91] A. Vahdat, B. Gao, M. Ranjbar, and G. Mori. A discriminative key pose sequence model for recognizing human interactions. In *ICCVW*, 2011.

[92] J. C. van Gemert, M. Jain, E. Gati, and C. G. Snoek. Apt: Action localization psroposals from dense trajectories. In *BMVC*, volume 2, page 4, 2015.

[93] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, 2013.

[94] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

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

[96] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. *arXiv preprint arXiv:1505.04868*, 2015.

[97] T. Wang, S. Wang, and X. Ding. Detecting human action as the spatio-temporal tube of maximum mutual information. *IEEE TCSVT*, 24(2), 2014.

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

[99] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[100] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars, 2007. ICCV.

[101] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *CVIU*, 115(2), 2011.

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

[103] J. Wu, F. Chen, and D. Hu. Human interaction recognition by spatial structure models. In *International Conference on Intelligent Science and Big Data Engineering*, 2013.

[104] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In *CVPR*, 2011.

[105] B. Xiaohan Nie, C. Xiong, and S.-C. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015.

[106] 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.

[107] R. Xu, P. Agarwal, S. Kumar, V. N. Krovi, and J. J. Corso. Combining skeletal pose with local motion for human activity recognition. In *Articulated Motion and Deformable Objects*.

2012.

[108] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[109] 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.

[110] 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.

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

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

[113] G. Yu, J. Yuan, and Z. Liu. Predicting human activities using spatio-temporal structure of interest points. In *ACM MM*, 2012.

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

[115] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia. Motion context: A new representation for human action recognition. In *ECCV*. 2008.

[116] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng. Online human gesture recognition from motion data streams. In *ACM MM*, 2013.

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