STARS

University of Central Florida
STARS

Electronic Theses and Dissertations

2017

Weakly Labeled Action Recognition and Detection

Waqas Sultani
University of Central Florida

 Part of the Computer Sciences Commons
Find similar works at: https://stars.library.ucf.edu/etd
University of Central Florida Libraries http://library.ucf.edu

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation
Sultani, Waqas, "Weakly Labeled Action Recognition and Detection" (2017). *Electronic Theses and Dissertations*. 5513.
https://stars.library.ucf.edu/etd/5513

WEAKLY LABELED ACTION RECOGNITION AND DETECTION

by

WAQAS SULTANI
B.E. University of Engineering and Technology, Taxila, Pakistan, 2006
MS. Seoul National University, South Korea, 2010

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

Summer Term
2017

Major Professor: Mubarak Shah

© 2017 Waqas Sultani

ABSTRACT

Research in human action recognition strives to develop increasingly generalized methods that are robust to intra-class variability and inter-class ambiguity. Recent years have seen tremendous strides in improving recognition accuracy on ever larger and complex benchmark datasets, comprising realistic actions "in the wild" videos. Unfortunately, the all-encompassing, dense, global representations that bring about such improvements often benefit from the inherent characteristics, specific to datasets and classes, that do not necessarily reflect knowledge about the entity to be recognized. This results in specific models that perform well within datasets but generalize poorly. Furthermore, training of supervised action recognition and detection methods need several precise spatio-temporal manual annotations to achieve good recognition and detection accuracy. For instance, current deep learning architectures require millions of accurately annotated videos to learn robust action classifiers. However, these annotations are quite difficult to achieve.

In the first part of this dissertation, we explore the reasons for poor classifier performance when tested on novel datasets, and quantify the effect of scene backgrounds on action representations and recognition. We attempt to address the problem of recognizing human actions while training and testing on distinct datasets when test videos are neither labeled nor available during training. In this scenario, learning of a joint vocabulary, or domain transfer techniques are not applicable. We perform different types of partitioning of the GIST feature space for several datasets and compute measures of background scene complexity, as well as, for the extent to which scenes are helpful in action classification. We then propose a new process to obtain a measure of confidence in each pixel of the video being a foreground region using motion, appearance, and saliency together in a 3D-Markov Random Field (MRF) based framework. We also propose multiple ways to exploit the foreground confidence: to improve bag-of-words vocabulary, histogram representation of a video, and a novel histogram decomposition based representation and kernel.

The above-mentioned work provides probability of each pixel being belonging to the actor, however, it does not give the precise spatio-temporal location of the actor. Furthermore, above framework would require precise spatio-temporal manual annotations to train an action detector. However, manual annotations in videos are laborious, require several annotators and contain human biases. Therefore, in the second part of this dissertation, we propose a weakly labeled approach to automatically obtain spatio-temporal annotations of actors in action videos. We first obtain a large number of action proposals in each video. To capture a few most representative action proposals in each video and evade processing thousands of them, we rank them using optical flow and saliency in a 3D-MRF based framework and select a few proposals using MAP based proposal subset selection method. We demonstrate that this ranking preserves the high-quality action proposals. Several such proposals are generated for each video of the same action. Our next challenge is to iteratively select one proposal from each video so that all proposals are globally consistent. We formulate this as Generalized Maximum Clique Graph problem (GMCP) using shape, global and fine-grained similarity of proposals across the videos. The output of our method is the most action representative proposals from each video. Using our method can also annotate multiple instances of the same action in a video can also be annotated. Moreover, action detection experiments using annotations obtained by our method and several baselines demonstrate the superiority of our approach.

The above-mentioned annotation method uses multiple videos of the same action. Therefore, in the third part of this dissertation, we tackle the problem of spatio-temporal action localization in a video, without assuming the availability of multiple videos or any prior annotations. The action is localized by employing images downloaded from the Internet using action label. Given web images, we first dampen image noise using random walk and evade distracting backgrounds within images using image action proposals. Then, given a video, we generate multiple spatio-temporal action proposals. We suppress camera and background generated proposals by exploiting optical

flow gradients within proposals. To obtain the most action representative proposals, we propose to reconstruct action proposals in the video by leveraging the action proposals in images. Moreover, we preserve the temporal smoothness of the video and reconstruct all proposal bounding boxes jointly using the constraints that push the coefficients for each bounding box toward a common consensus, thus enforcing the coefficient similarity across multiple frames. We solve this optimization problem using the variant of two-metric projection algorithm. Finally, the video proposal that has the lowest reconstruction cost and is motion salient is used to localize the action. Our method is not only applicable to the trimmed videos, but it can also be used for action localization in untrimmed videos, which is a very challenging problem.

Finally, in the third part of this dissertation, we propose a novel approach to generate a few properly ranked action proposals from a large number of noisy proposals. The proposed approach begins with dividing each proposal into sub-proposals. We assume that the quality of proposal remains the same within each sub-proposal. We, then employ a graph optimization method to recombine the sub-proposals in all action proposals in a single video in order to optimally build new action proposals and rank them by the combined node and edge scores. For an untrimmed video, we first divide the video into shots and then make the above-mentioned graph within each shot. Our method generates a few ranked proposals that can be better than all the existing underlying proposals. Our experimental results validated that the properly ranked action proposals can significantly boost action detection results.

Our extensive experimental results on different challenging and realistic action datasets, comparisons with several competitive baselines and detailed analysis of each step of proposed methods validate the proposed ideas and frameworks.

.

ACKNOWLEDGMENTS

All praises are due to Allah, the only worthy of worship, the most gracious and the most merciful. I am grateful to Dr. Mubarak Shah for spending hundreds of hours over the past several years to make this dissertation possible. His discipline, commitment and hard work is a great source of inspiration for me. I would like thank to my all present and past colleagues in CRCV. I would also like to thank my friends who support me in difficult times especially: Imran Saleemi, Sarfaraz Hussein, Khurram Soomro, Fahd Khan, Sehar Butt, Muhammad Rehan and Mehboob-ur-Rehman among others. I would like to thank my Ph.D committee members Dr. Ulas Bagci, Dr. Hae-Bum Yun and Dr. Guo-Jun Qi for their valuable comments. Finally, I would like thank my father Dr. Muhammad Ibrahim Sultani and my mother and wife for their countless prayers and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Humans are social species and have the ability to recognize actions of other humans. Humans can detect actions irrespective of who is performing actions as well as recognize actors irrespective of actions they are performing. In computer vision, we want to develop the same capability of action recognition for computers. Specifically, we want computers that can recognize complex human actions and different actors in various indoor and outdoor environments using videos obtained by hand-held or surveillance cameras. The development of such a robust action recognition system has several practical applications and can have the significant impact on our daily lives. Some of its applications are: (1) It can enhance human-computer interaction and let robots do their tasks automatically by properly responding to human actions; (2) In surveillance applications, it can automatically detect suspicious and abnormal activities such as crimes and illegal activities and has potential to save several lives; (3) It can facilitate detailed analysis of sports videos and can help athletes to further improve their skills; (4) It can alleviate some of the problems in text-based video indexing and retrieval and improve it through recognizing activities within videos; (5) finally, it can develop intelligent environment systems to facilitate taking care of elderly people.

In *action recognition*, the main goal is to classify whether the testing video clip contains the specific action or not, irrespective of the location of actors. Predicting the spatio-temporal location of a known action in the video is termed *action localization*. In several applications, in addition to recognition, we also want to estimate the precise spatio-temporal locations of actors. This is termed *action detection*. In Figure 1.1, we show an example of action recognition and detection.

Although action recognition and detection are two of the most important computer vision tasks, they are still far from being solved, mainly due to large variations of action appearance and complex human poses, variable scales and viewpoints, cluttered backgrounds in crowded scenes and

different illumination conditions.

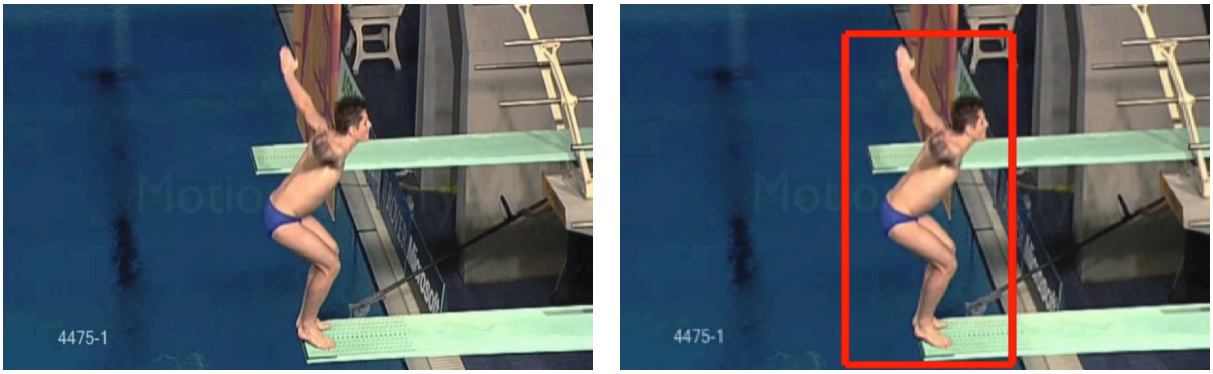

Figure 1.1: In action recognition, the goal is to detect the action ('Diving' in this figure) in the video, irrespective of location of actor (Left figure). Right figure shows an example of action detection, where in addition to correctly classifying the lable of video, we need precise locate spatio-temporal position of the actor.

Most of the descriptors and detectors for action recognition and detection are derived from that of image classification and object detection techniques. In addition to spatial information, action recognition descriptors contain temporal information. In comparison to images, videos contain much more noise, complex articulations of objects and have huge search space. For instance, in a typical exhaustive object detection approach, we need to search $O(w^2 \times h^2)$ locations to successfully detection an object, where w and h represent width and height of an image. On the other hand, for detecting an action in a video of size $w \times h \times F$ where w, h and F represent width and height of a video frame and F represents total number of frames, we need to search $O(w^2 \times h^2 \times F^2)$ locations which is much more computationally expensive as compared to image search space. These problems motivate action recognition research towards foreground focused representations. The ideal foreground representations should be derived from actor only, but attempting to estimate actor bounding boxes or binary foreground-background labels is akin to introducing a new problem to solve the first. Since the purpose of foreground representation is to facilitate action detection and

reduce noise, the approach to estimate foreground regions should be fast and needs as minimum supervision as possible.

A conventional approach to training an action detector is to first collect several examples of the action (also called positive examples) as well as examples which do not contain action of interest (also called negative examples). The approach that use accurately collected positive and negative examples of training is called *supervised approach*. These supervised approaches have attained state-of-the-art accuracy for detecting human action in videos; however, their dependence on precise annotations make them costly to use. The challenges in obtaining manual annotations lead to the development of approaches which need minimum or no supervision. In weakly supervised approaches, a classifier does not need complete annotations about the concept to be learned. For instance, for training a spatio-temporal action detector, videos which contain action of interest can be used without providing a precise spatio-temporal location of an actor. Note that weakly supervised approaches have been successfully used in action and object detections. In the unsupervised method, even video level labels are not available. Given the pool of videos, we want to automatically learn action detector for all actions happening in the videos. These methods do not require any annotations. Although very useful, their accuracy is still far below from supervised methods.

In this dissertation, we aim to address the problem of action recognition and detection in weakly labeled videos. In weakly labeled videos, we only have video level labels (defining whether the video contains a specific action or not) available instead of precise pixel-wise or bounding box annotations, neither during training nor during testing. We make following important contributions in this dissertation.

- We propose a new approach to obtain foreground focused representation for action recognition and demonstrate its applications in recognizing human actions while training and testing on distinct datasets when test videos are neither labeled nor available during training. We

investigate the reasons for poor classifier performance when tested on novel datasets, and quantify the effect of scene backgrounds on action representations and recognition. We present a new process to obtain a measure of confidence in each pixel of the video being a foreground region, using motion, appearance, and saliency together in a 3D-MRF based framework. We also propose multiple ways to exploit the foreground confidence: to improve bag-of-words vocabulary, histogram representation of a video, and a novel histogram decomposition based representation and kernel.

- We present a weakly labeled approach to automatically obtain spatio-temporal annotations of actors in action videos. Here, we exploit the foreground representation to select a few proposals using MAP based proposal subset selection method. Next, we iteratively select one proposal from each video so that all proposals are globally consistent. We formulate this as Generalized Maximum Clique Graph problem using shape, global and fine-grained similarities of proposals across multiple videos. The output of our method is the most action representative proposals from each video. Experimental results on several challenging datasets and competitive baselines demonstrate the superiority of proposed approach.

- We present a novel approach for action localization using Web images. We neither assume the availability of multiple videos or any prior annotations. The action is localized by employing images downloaded from Google images using action label. We propose to reconstruct action proposals in the video by leveraging the action proposals in images. Moreover, we preserve the temporal smoothness of the video and reconstruct all proposal bounding boxes jointly using the constraints that push the coefficients for each bounding box toward a common consensus, thus enforcing the coefficient similarity across multiple frames. We solve this optimization problem using a variant of two-metric projection algorithm.

- We propose a new approach to generate a few better action proposals that are ranked properly. In this approach, we first divide action proposal into sub-proposal and then use Dynamic

Programming based graph optimization scheme to select the optimal combinations of sub-proposals from different proposals and assign each new proposal a score. We propose a new unsupervised image-based actioness detector that leverages web images and employ it as one of the node scores in our graph formulation. Moreover, we capture motion information by estimating the number of motion contours within each action proposal patch. We experimentally demonstrate that properly ranked proposals produce significantly better action detection as compared to state-of-the-art proposal based methods.

1.1 Foreground Focused Action Recognition

Although background environment and contextual information of an action is important and should be taken into consideration, its contribution to the final representation should be less than the action itself. The representation of an action should be actor-centric so that a classifier learns the action and not the backgrounds and hence is able to recognize actions in a different backgrounds.

While obtaining meaningful interest points, actor contours or silhouettes in modern action datasets is challenging, we nevertheless, argue that a truly representative action model that generalizes reasonably well across unseen datasets, would benefit from the same cues that are used for foreground segmentation. Since estimation of actor bounding boxes or binary foreground-background labels is difficult, we propose to perform *unsupervised* estimation of pixel-wise real-valued labels for the entire video that can be employed to control the influence of different video regions on the final representation.

We put forth three methods to exploit these foreground confidences for soft assignment of features within the bag-of-words paradigm. The two main aspects of our proposed methods are: important features should have a larger influence in video representation, and regions with a specific level of

importance should only be compared with corresponding similarly important parts of other videos.

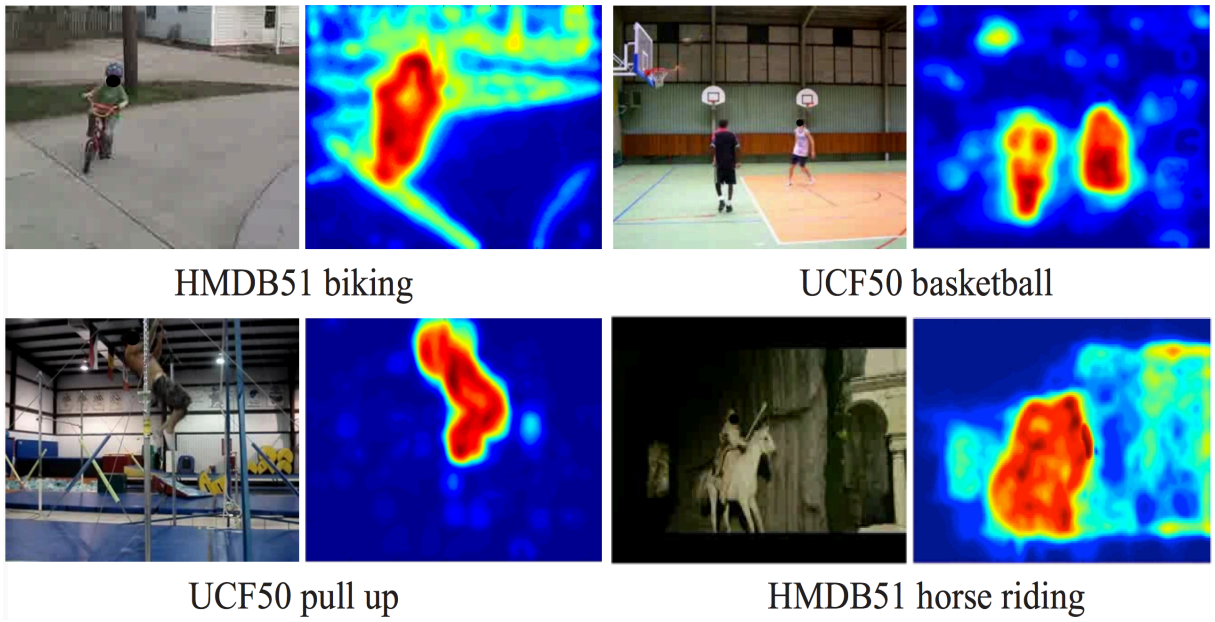

Figure 1.2: Original video frame, and corresponding confidence of each pixel being the foreground for four example videos.

In our first representation, weighted bag-of-words, each word contributes to the histogram according to its probability of being foreground. In our second representation, foreground based histogram decomposition, we divide video into arbitrary (potentially *non-contiguous*) regions according to their probability of being foreground, and the final distance between two videos is the summation of *weighted* distances between regions that correspond to the same quantized probability of being foreground. In fact, we are not aware of any previous methods attempting the problem of action recognition across datasets, that do not exploit either labeled or unlabeled videos (or features) from the test dataset. In Figure 1.2, we demonstrate our unsupervised estimation of foreground regions in action videos from two different datasets.

The aim of action recognition research is to design a robust system which can detect action accurately in real world scenarios. To evaluate such system, several benchmark datasets have been

collected [37, 67]. Recent action recognition approaches [87, 85] perform quite well on these datasets. However, such improvements often benefit from the inherent characteristics, specific to datasets and classes, and learn the dataset instead of actions. Such classifiers work quite well when tested on the training dataset, however perform miserably poor when the testing set is from different training action dataset.

Similarly, conventional *detectors* learn classifiers from labeled examples and assume that the videos to be tested belong to the same distribution(s) as the training set. However, this might not be the case when testing videos belong to a different dataset. The need to mitigate this disconnect has given rise to the application of domain adaptation [4, 12], in recognition of objects [56] and events [13, 92]

There is no question that these techniques improve performance across datasets, and are significant in their own right, but it is worth asking whether the same actions in distinct datasets are truly representative of different domains or if their specific characteristics are distracting biases that emanate from data collection criteria and processes. This issue has been raised recently in an interesting work by Torralba and Efros [77] for the problem of image classification and object detection. They have empirically established that most object recognition datasets represent close visual world views and have biases toward specific poses, backgrounds, and locations, etc. In the analysis given in the third chapter of this dissertation, we show that action recognition datasets too are prejudiced towards background scenes – a characteristic that should ideally be inconsequential to human action classes.

Explicit mitigation of dataset bias is possible, we argue, however, that research should focus on video (or image) *representation* instead, so it is invariant to said bias and potentially generalizes better across datasets. The underlying assumption is that the hypothetical, exhaustive set of examples of an action class would be truly representative of our visual world, and treating distinct

datasets as training and testing partitions is a step towards realizing such a set. We propose that dataset invariant action representations should attempt to capture features of the actor's motion and appearance along with involved objects, and diminish the effect of scene background and clutter. This way action classifiers would learn the action and not the background and hence expected to classify actions correctly with different background scenes.

Historically, taking a page from image analysis, several *video* interest point detectors were introduced, including space-time interest points [43], Dollar interest points [11], and spatiotemporal Hessian detector [90], etc. The obvious idea was to estimate local descriptors only at these important locations and ignore the rest of the video. Representations based on local descriptors estimated at interest points showed promising results on simple datasets such as Weizmann [60] and KTH [58]. Even though these datasets are now considered easier, their generally static, mostly uniform scene backgrounds, coupled with interest point detection, ensured a true action representation, largely devoid of background information.

In recent years, the difficulty in obtaining meaningful locations of interest in contemporary datasets, coupled with the lack of evaluation of action localization, has resulted in a shift in research focus away from interest point detection. Indeed, it has been shown experimentally, that dense sampling of feature descriptors generally outperforms interest point [87] and other detectors (human, foreground)[35]. Several methods have even been proposed to recognize actions in single images instead of videos. It is then safe to assume that background scene information is a key component of the final representation that allows higher quantitative performance, but in the process 'learns the dataset' rather than the action. Therefore, it is not surprising that the methods that detect actions with more than 90% accuracy in trimmed dataset fail to detect the same action when happening in long untrimmed videos.

We maintain that the goal of action representation schemes and efforts to collect larger datasets

should be to increase intra-class generalization for which cross-dataset recognition is a reasonable metric.

1.2 Automatic Action Annotations

Although estimation of foreground regions can reduce search space in videos and improve classification accuracy, we still need a large number of precise spatio-temporal annotations to train a robust action detector.

The spatio-temporal annotations are cumbersome to obtain, require many human annotators, hundreds of hours, expensive annotation interfaces and are subject to human biases. Moreover, for any new action class, the annotation needs to be done from scratch. As action datasets are exponentially growing, design and development of generic automatic annotation methods are very much needed. This does not only reduce human biases but also saves time and cost.

With the advent of large image and object datasets [55], automatic object annotation is becoming challenging. Hence, it's gaining more attention from the research community. To address this issue, several approaches have been presented recently to obtain object level annotations from image level annotations. These approaches automatically obtain object bounding box location using: eye-tracking [50], transferring annotations from previously annotated object to the new class [23], exploiting generic object knowledge [10], jointly localizing objects in multiple images [51, 72] and using videos [32]. The straightforward extension of these approaches to automatically obtain action spatio-temporal annotations from video level labels is not feasible because temporal domain is quite different from spatial domain. Temporal length of an action can be arbitrarily long depending upon action cycles captured in a video. In addition, 3D cuboids would contain significant background pixels due to large camera motion and spatial motion of an actor.

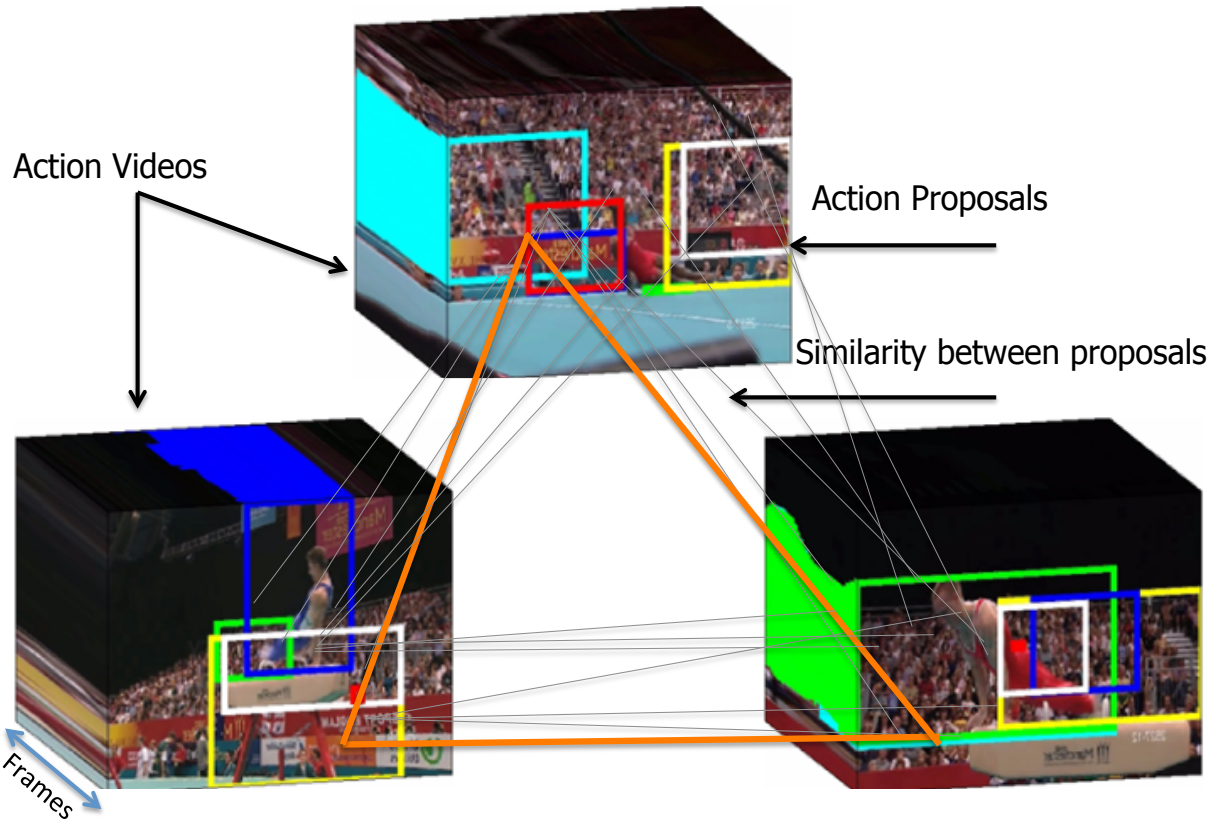

Figure 1.3: We build a fully connected graph across top ranked proposals in multiple videos of the same action. The proposals which are consistent across videos (orange graph) are selected as action annotations.

Instead of 3D cuboids, spatio-temporal action proposals obtained through segmentation [91] or dense trajectories [81] of a video can precisely enclose action boundaries and capture arbitrary spatio-temporal action localization. We believe that the action proposals can provide a useful platform to obtain automatic action annotations in videos. However, these methods produce a humongous number of action proposals; some precisely enclose the complete action, while the majority are noisy and capture only action parts, background, camera motion, or both foregrounds-backgrounds.

In chapter 4, we present a new approach to automatically discover the most action representative

proposals from each action video that tightly covers actor spatio-temporal localization. We propose to obtain these spatio-temporal action annotations using videos level labels. Given action proposals, we seek to discover automatically the proposals that have the higher probability of representing the spatio-temporal location of an actor. Given a large number of proposals, we initially rank them according to their probabilities of being representative of an action. We achieve this using Maximum-a-Posteriori (MAP) based subset selection procedure by employing optical flow gradients and saliency in the 3D-MRF based framework.

We then utilize similarity between top ranked proposals across different videos of the same action and re-rank top proposals (see Figure 1.3). For this purpose, we build a fully connected graph where all proposals in one video are connected to every proposal in all other videos and the edges between proposals capture global, fine-grained and shape similarities between proposals. Finally, we formulate the proposals matching across multiple videos as a Generalized Maximum Clique Problem (GMCP). The proposals that form maximum clique are used as action annotation.

Our method is weakly labeled as we only use video level labels instead of bounding box level annotations. It is efficient since for a typical action datasets such as UCF-Sports and sub-JHMDB, we achieve the final bounding box action annotations within a few seconds using GMCP employing only a few top ranked action proposals. Our approach is useful, as it can seamlessly be integrated with any other action detection method. With these key aspects, our method satisfies three main characteristics of a visual system: less supervision, efficiency, and usefulness.

1.3 Action Localization Using Web Images

Although the above-mentioned annotation method works reasonably well, it has some main limitations: (1) It requires the availability of multiple videos of the same action, which is not always easy

to obtain, (2) Processing of multiple videos requires significant processing and memory resources. Therefore, the better system would be the one that can annotate single video independently. This will require fewer memory resources and allow us to benefit from recent parallel processing techniques such as GPUs to localize all videos at the same time in parallel.

To this end, we present a new approach in chapter 5 to tackle the challenge of action localization in a single video. We propose to leverage images downloaded from the Internet using text-based queries. In contrast to previous works in object annotations, we neither assume the availability of bounding box annotations nor the presence of multiple videos of the same class. Furthermore, we do not assume the availability of clean images either.

Images are usually taken to capture key poses, descriptive viewpoints and important instances of an action or event. Our key idea is to exploit this useful information to obtain precise spatio-temporal action localization in *videos*. To operationalize our intuition, we first download several images of the action of interest using the action label as a query from Google. These images contain human performing actions in different locations (not necessarily at the center), backgrounds and include many irrelevant and noisy images. To circumvent these issues, we remove irrelevant noisy images using random walk. To handle the challenge of variable locations and backgrounds, we co-localize the common action in multiple images. The output of these steps is candidate action localization in the images.

Our ultimate goal is to obtain spatio-temporal annotations in a video. Therefore, given a video clip, we first obtain action proposals [49]. Action proposals represent candidate spatio-temporal action locations in the video that might contain an action. However, not all proposals are truly action representative as many are due to camera motion and cluttered backgrounds. Therefore, we remove highly overlapping action proposals using non-maximal suppression by exploiting optical flow gradient within the proposals. To obtain the most action representative proposal, we propose

to reconstruct action proposals in the video by leveraging the action proposals in images.

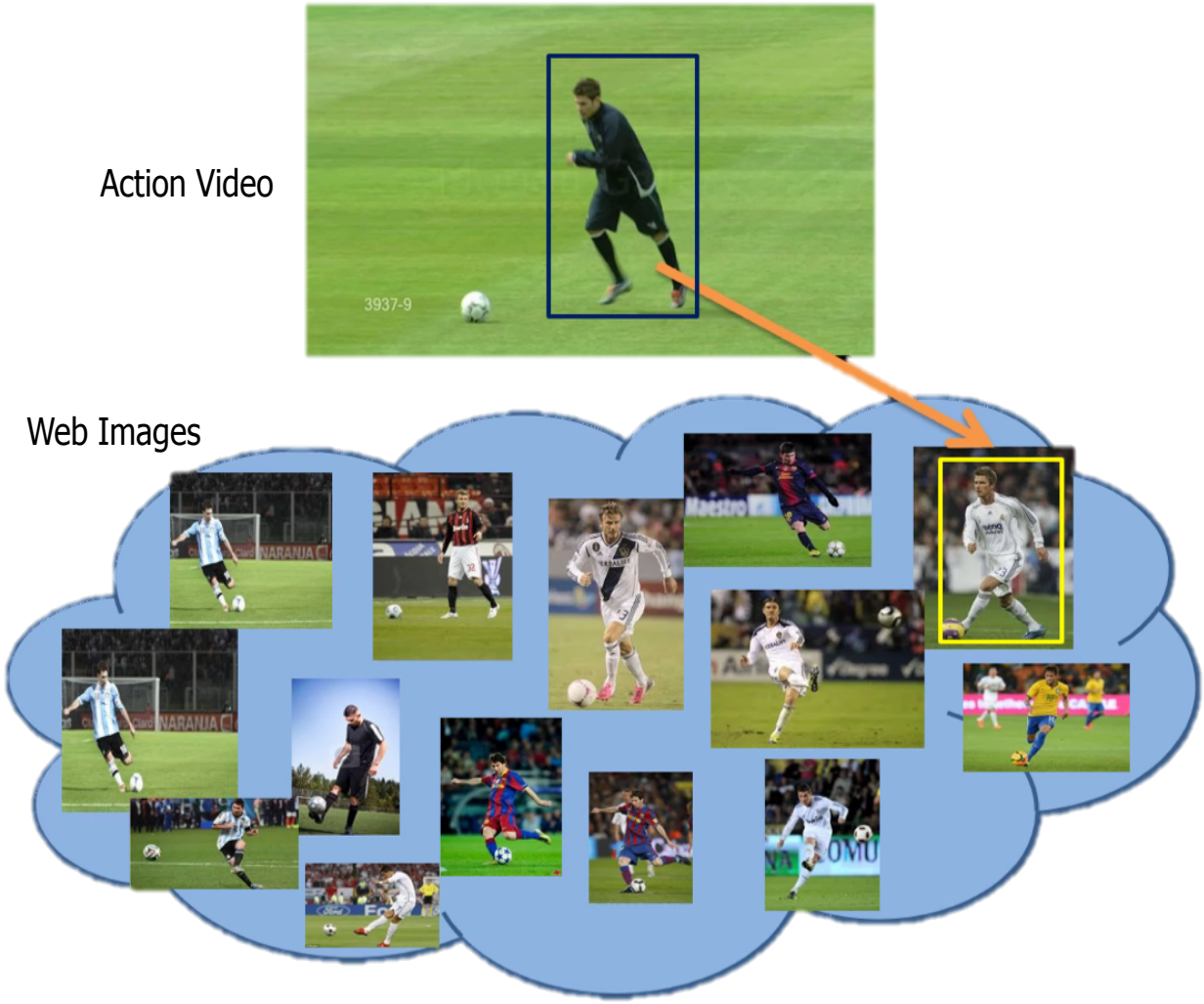

Figure 1.4: Image capture key poses, descriptive viewpoints and important instances of an action or event. Our idea to leverage these to achieve spatio-temporal localization in videos.

Furthermore, we preserve the temporal smoothness of the video by introducing consensus regularization. Consensus regularization enforces consistency among coefficients vectors of multiple frames within the proposal. The proposal with the lowest reconstruction error and a high motion saliency is selected as a final action localization. Figure 1.4 illustrates the proposed idea.

Our experimental results reveal that it is possible to automatically annotate an action in a video by employing web images of the same action through mitigating the effect of distracting backgrounds within images and by preserving the temporal structure of video during reconstruction.

Most of the previous works demonstrate action localization accuracy either on trimmed videos or carefully staged clean untrimmed videos. However, these videos do not represent the real-world videos, which are long, have variable scenes and backgrounds and contain multiple or no instance of the action of interest. Since proposed approach does not require multiple videos and prior annotations, it can easily be applied to more realistic untrimmed videos. We evaluate our approach on trimmed [54] as well as on the part of untrimmed [31] datasets and obtain encouraging results.

In summary, (1) We demonstrate the feasibility of using images to achieve spatio-temporal action localization in videos, (2) By utilizing video proposal sparse reconstruction error with motion saliency, we achieve impressive localization results on popular trimmed action dataset, (3) We are the first to report spatio-temporal action localization results on (the part of) challenging untrimmed action dataset THUMOS14 [31]. Furthermore, we release spatio-temporal annotations of 35, 000 frames of THUMOS14 to facilitate further research in this direction.

1.4 Action Proposal Ranking Through Proposal Recombination

Recently, several action proposal methods have been introduced to reduce the search space and improve action localization accuracy. These proposal methods use hierarchical segmentation or clustering and therefore, they tend to generate thousands of action proposals in each video, where many of proposals are noisy and do not contain action. Moreover, these methods utilize low-level color, motion and saliency cues but ignore high-level action cues, which make them difficult to generate an actioness score for each proposal. Hence, classification methods treat all propos-

als equally. Fully supervised action proposal methods such as [96] employ thousands of manual human annotations and use motion information from training videos to obtain ranked action proposals. However, with the recent explosive growth of action datasets [31], it is prohibitive to obtain bounding box annotations for each video, thus the application of fully supervised action proposal methods is limited.

In chapter 6, we address the above-mentioned limitations of action proposals methods. Given the output of any recent action proposal method, our goal is to generate a few new action proposals which are properly ranked according to how well they localize an action (see Figure 1.5). The proposed method assigns an accurate actioness score to each proposal, without using any labeled data (no human detection or bounding box annotations), thus is easier to generalize to other datasets.

The proposed approach begins with dividing each proposal into sub-proposals. We assume that the quality of proposal remains the same within each sub-proposal. We, then employ a graph optimization method to recombine the sub-proposals in all action proposals in a single video in order to optimally build new action proposals and rank them by the combined node and edge scores. The node score is a combination of an image-based actioness score (or image actioness; since this is computed for an image, not a video) and motion scores. The image-based actionnes scores are obtained by a deep network trained on web images. The training data is obtained without manual annotation by an automatic image co-localization method. In essence, the trained deep network is a generic image actioness detector rather than a specific action detector. The motion scores are obtained by measuring the number of motion contours enclosed by each proposal bounding box patch. The edge scores between sub-proposals are computed by the overlap ratio and appearance similarity of temporally neighboring sub-proposals. Note that edges are made between temporally adjacent sub-proposals. For an untrimmed video, we first divide the video into shots and then make the above-mentioned graph within each shot. Our method generates a few ranked proposals that can be better than all the existing underlying proposals. Our experimental results validated that the

properly ranked action proposals can significantly boost action detection results.

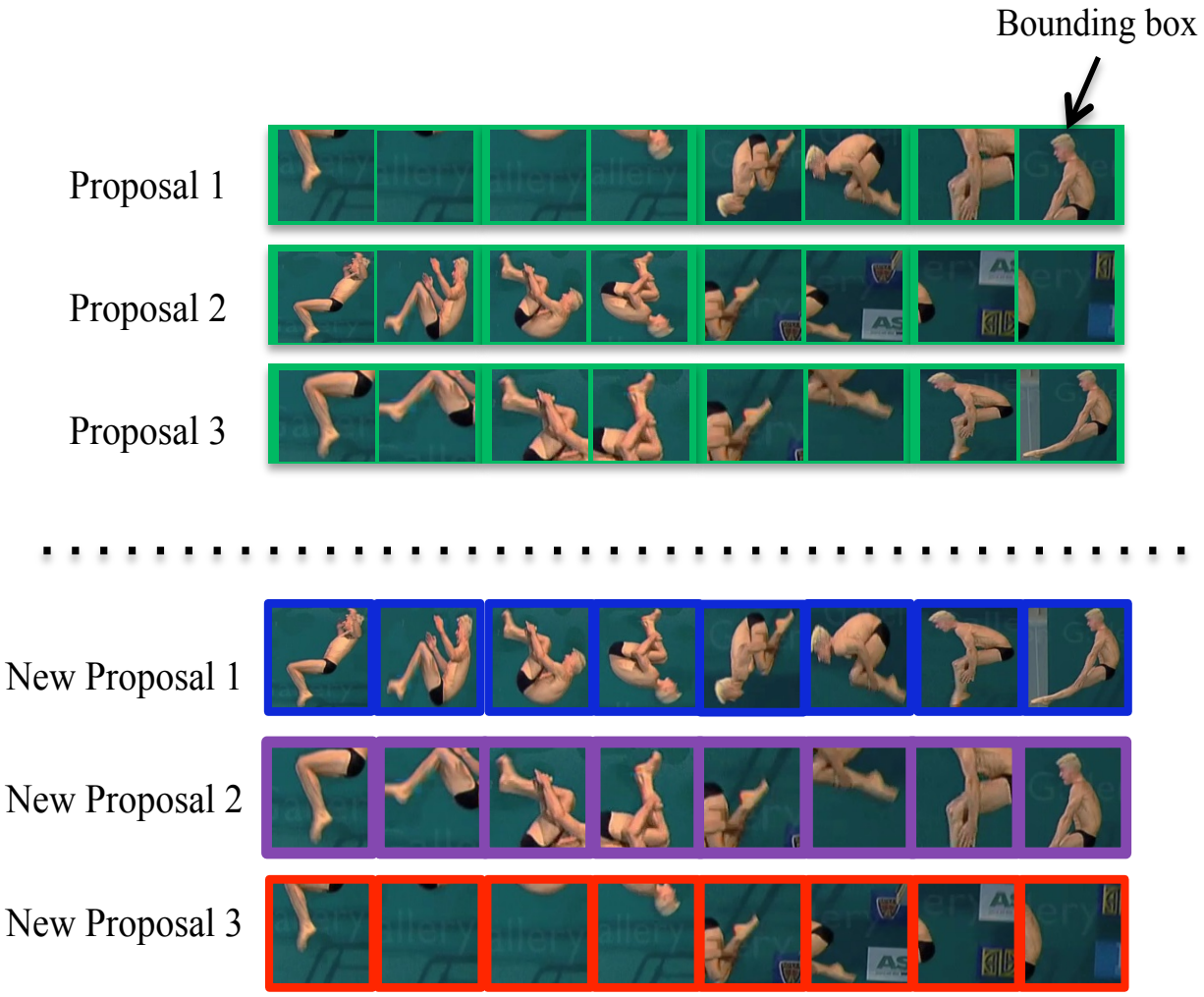

Figure 1.5: Top: we show three noisy action proposals. Employing these noisy action proposals, we want to obtain new good quality action proposals which are ranked properly (shown at the bottom), where New Proposal 1 shows new top ranked proposal and New Proposal 3 represents new lowest-ranked proposal.

1.5 Dissertation Organization

This dissertation is organized as follows. In Chapter 2, an overview of recent supervised, weakly action detection methods, automatic annotation approaches, and techniques that used image and videos jointly is provided. In Chapter 3, we provide analysis of discriminative effects of action backgrounds and present a new method for cross-dataset action recognition. In Chapter 4, we present our weakly labeled approach for action annotations. Chapter 5 describes our action localization technique using Web images. Chapter 6 introduce unsupervised action proposal ranking approach. Finally, in Chapter 7 we present conclusions and possible future directions.

CHAPTER 2: LITERATURE REVIEW

Action recognition and detection are two of the long-standing unsolved problems in Computer Vision. Several methods have been presented for last two decades including interest point detection approaches, dense feature sampling and recently introduced deep learning approaches. Overall, these approaches extract visual features from videos, make a compact representation of all the features in each video and learn classifier discriminate different actions.

Due to the difficulty of the problem, the majority of action recognition and detection research employ fully supervised methods, where the complete manual annotations are assumed. Since in action recognition the spatio-temporal localization is not required, supervised approaches only need videos level annotations. However, for action detection, which also provides spatio-temporal localization, precise spatio-temporal annotations are needed as well. Note that most of the supervised action recognition approaches rely on dense features and ignore location of the actor. Due to a large number of background features, sometimes dense representation is not discriminative. Moreover, dense feature approaches are susceptible to dataset bais. In order to improve the accuracy of recognition and reduce the effect of backgrounds, several foregrounds focused representations have been introduced recently. These approaches use visual saliency-based features or divide the video into different regions.

Supervised methods need manual annotations, which are laborious to obtain. For instance, recent deep learning frameworks need millions of videos to learn robust representation. This motivates the development of weakly supervised approaches that do not require complete labels to learn the action model. Using only video level labels, weakly supervised action recognition approaches automatically discover discriminative regions (not necessarily actor location) in videos to boost action recognition accuracy; whereas weakly supervised action detection approaches estimate pre-

cise action location of the actor in addition to recognition.

In section 2.1, we briefly introduce some of the recent fully supervised action recognition and detection works. We describe foreground-focused methods in section 2.1.1. In section 2.2, we review recently proposed weakly supervised methods for action recognition and detection. In section 2.3, we present latest methods that localize and segment objects jointly in multiple videos and images. Finally, in section 2.4, we discuss approaches which utilize image and videos jointly for different computer vision applications.

2.1 Supervised Action Recognition and Detection in Videos

Inspired by recognition capability of human brain, researchers proposed to extract different color and edge based visual features from images. These visual features capture the key information from images and are used to discriminate images of different classes. Furthermore, in order to extract discriminative visual features, several interest point detector were introduced. Following the success of interest point detectors for image classification, several spatio-temporal interest point detectors were introduced for action classification in videos. These interest point detection methods detect interesting points in videos and extract features from those interesting points. In 2009, Wang et al. [87] demonstrated that features extracted at dense locations outperform interest point detector approaches by the significant margin. This work is followed by dense and improved dense trajectory features [84, 86]. Improved dense trajectory approach track thousands of corner points in the video and extracts HOG, HOG, MBH and Trajectory displacement features along the trajectories. Combined with Bag of Word (BoW) framework, this approach outperforms previous state-of-the-arts by a large margin.

Inspired by the success of deep learning features for image classification, several deep learning

approaches have been presented for video action classification. Tran et al. [78] recently presented an efficient video descriptor by employing 3D convolution. Although impressive; their recognition accuracy is still less than the hand-crafted feature approaches. Moreover, due to the complexity of deep architecture, they trained their models on one million YouTube videos. Simonyan et al. [62], reported a 2-stream network for action recognition. Their proposed architecture include spatial network, trained on RGB images, and temporal network, trained on the dense optical flow field. Similarly, inspired by deep learning success in object detection, Gkioxari et al. [21] proposed to use two stream network using object proposals for action localization. Note that due to the lack of sufficiently annotated video data, they used models which were trained on millions of images and were fine-tuned on action datasets. Instead of detecting action in the whole video, they first detect action in each frame of the video. After that, they link the high scored action boxes across frames to compute action tubes. Weinzaepfel et al. [89] presented approach similar to [21] and employed tracking by detection approach to obtain action localization. Again, due to the lack enough video annotations, they used models trained on images. Furthermore, they have to use hand-crafted features along with CNN features to achieve stat-of-the-art action detection accuracy.

2.1.1 Foreground-Focused Action Recognition

Due to difficulty in obtaining meaningful locations of interest points and lower performance of human detectors in realistic action datasets such as THUMOS13, researchers have designed dense sampling methods and have shown much better performance for action recognition task as compared to detecting human and foreground. However, in parallel, some researchers have continued to design foreground focused object and action recognition methods. Authors in [82] proposed the novel soft assignment of features instead of traditional hard quantization for histogram formation in the famous Bag of Words (BoW) framework. Indeed, they demonstrated significantly improved accuracy as compared to BoW. In order to incorporate context information in histograms, context

specific histograms were proposed in [68] for image classification, where different words contribute differently to each histogram. However, the context classifiers need to be pre-trained in a supervised manner. Ullah *et al.* [80] integrated region level information in BoW framework, where region level information was obtained through segmentation, person detection and static action and object detection. Vig et al. [26] used saliency to find foreground regions in the video and use saliency information to change the histogram formation. Specifically, they remove features from non-salient regions by thresholding the saliency map and each of the remaining features contribute equally to the final representation. Moreover, in their second representation, they assign more weights to the features originating from salient regions using additional codebook vectors.

Unlike the above foreground-focused representations, in the first foreground-focused representation that we propose in this dissertation, each word contributes to the histogram according to its probability of being foreground. In our second representation, we divide the video into arbitrary regions according to their probability of being foreground, and the final distance between two videos is the summation of weighted distances between regions that correspond to the same quantized probability of being foreground. Moreover, our motivation and application of foreground representation are different from the above-mentioned papers.

The above-mentioned fully supervised methods need hundreds of spatio-temporal annotations. To the best of our knowledge, we are unaware of any stat-of-the-art deep action detection system which is using videos only. As mentioned in section 2.1, current deep action detection methods used models trained on images due to the lack of sufficiently annotated video data. These limitations lead to the developments of several weakly supervised approaches, which are discussed next.

2.2 Weakly Supervised Action Recognition and Detection

With the exponential increase in the size of object/action datasets, obtaining annotations is becoming an increasingly daunting task. Moreover, it is subject to human biases in terms of temporal start and end of the activity and the sizes of the exact spatial boxes around an actor.

Most of action localization approaches are inspired by object localization, extending the detection problem from 2D bounding box to the 3D cuboid. Due to a fixed size of the cuboid, detection results of these methods contain a significant portion of the background, especially, when actor's aspect ratio varies significantly. To circumvent this problem, more precise action detection approaches have been introduced [27, 40, 81]. However, training action classifiers using any of the above methods requires hundreds of time-consuming spatio-temporal annotations. In order to avoid these time-consuming annotations, weakly supervised methods have been introduced recently for training action classifiers [3, 59, 64]. These methods require only video level labels. Boyraz et al. [3] presented a weakly supervised action recognition method to estimate discriminative regions in each frame. The histograms of these discriminative regions are used for learning the action classifier using two-layer neural networks. Similarly, authors in [59] reported a method where discriminative regions are considered as latent variables. They proposed a similarity constrained latent SVM, which jointly learns the action classifier as well as discovers discriminative regions. Both of these methods have shown improved classification accuracy using discriminative regions without requiring manual spatio-temporal annotations. Although they provide improved results, as mentioned in the papers [3, 59], automatically discovered discriminative regions do not necessarily represent human action locations. The authors in [64] proposed a weakly supervised action detection method based on multiple instance learning. However, one of the major limitations of their method is the assumption that actions can only be performed by standing persons. Therefore, their method is not suitable to recent action datasets [25, 37, 54] which contain huge

articulated human motion.

Overall, these weakly supervised action recognition and detection approaches use image/video level labels only and learn the robust object/action classifier and detector without requiring bounding box annotations. Although impressive, their accuracy is still far behind than that of detectors trained on hundreds of manual bounding box annotations.

2.3 Co-localization

Most of the action recognition works that we have discussed so far obtain action detection using a single video at test time. In this standard setting, action in each video is detected independently without considering other testing videos. However, it would be more robust to infer action location using multiple videos jointly. In this way, a better localization can be achieved by exploiting the similarity across multiple videos. Since this type of localization is achieved using multiple videos; it is called *colocalization*. Note that co-localization has been used in supervised, weakly supervised as well as unsupervised approaches.

Untill the last few years, most object detection methods used multi-scale sliding window approaches. In order to reduce search space and computation complexity of sliding window approach, several improvements have been made [79, 41]. The most popular approach in this direction is to use object proposals [1, 45, 79, 101]. The object proposal provides category independent candidate object location which may contain an object. Due to their high recall and efficiency, it is not surprising that most of the successful object detection approaches use object proposals as their pre-processing steps [20].

Following selective search based object proposal method [79] in images, Jain et al. [27] presented a video based action proposal method using motion information. The method [27] starts with

supervoxel segmentation [91], followed by a hierarchical grouping method using motion, color and texture cues. The method shows improved action classification results compared to several sliding window alternatives. Similarly, Oneata et al. [49] extended [45] from images to videos and introduced randomized supervoxel segmentation method for proposal generation. Although both methods [49, 27] produce good quality proposals, they are computationally expensive due to their dependencies on supervoxel segmentation. Recently, [81, 96] presented a faster method for generating action proposals by removing supervoxel segmentation altogether. Yu el al. [96] generated action proposals using the supervised human detector and connecting human detection boxes across multiple frames. In addition to action proposals, authors in [96] also produce a probability of actionness for each proposal. Van Gemert et al. in [81] employ dense trajectory features [86] and produce accurate action proposals by clustering [61] dense trajectory features.

Although object and action proposals are mainly designed to reduce search space and improve the efficiency of object and action classification; they have also been used to obtain automatic object localization in videos and images. Below we discuss few prominent techniques, which have been introduced recently for object localization in images and videos.

2.3.1 Co-localization in Images

Co-localization in images employs multiple images jointly to infer the location of a common object. In weakly labeled colocalization, all images use in colocalization are assumed to have a dominant common object. Tang $et\ al.$ [73] introduced co-localization method where the objective is to obtain bounding boxes around common objects among multiple images. They first extract several object proposals in each image and then compute several image features within those bounding boxes to capture appearance information. In their formulation, they used both images and box features to obtain common object co-localization. Although their joint image and box formulation

can also handle the presence of noisy images to some extent, it is weakly supervised method since it assumes the availability of image level labels.

Recently, unsupervised colocalization has been proposed by Cho $et\ al.$ [8]. They introduced an unsupervised part based matching approach to localize common objects across multiple images, without requiring images level labels. They extract segmentation based object proposals from each image and proposed a new probabilistic hough matching along with a new salient region detection method to find the discriminative objects and their parts in each image. They also use global GIST features to discover similar images for matching. Given several images of different object classes, this method efficiently localizes objects which are common in multiple images.

2.3.2 Co-localization in Videos

Colocalization has also been introduced for joint object localization in videos. Similar to weakly labeled colocalization in images, weakly labeled colocalization in videos assume common object in all videos. Prest et al. [51] proposed to use motion segmentation and obtain several spatio-temporal tubes from multiple videos containing the same target object. After that, they select one tube per video using all the videos jointly. The final selected tubes contain the object in different frames of videos and can be used to augment the training data. Although this approach works quite well, however, it has the underlying assumption of clean videos.

Jounin et al. [32] extended the approach discussed in [73] from images to videos and co-localize objects in several frames using multiple videos. After extracting several object proposals in each image, they track proposals using motion consistency information across frames of videos. They propose an efficient formulation of Frank-Wolfe algorithm to find common pattern across images and videos. Furthermore, they also presented quantitative results for joint image and video colocalization.

Although encouraging results have been obtained, the above-mentioned methods require multiple videos or images of the same object of interest and cannot localize the objects if multiple videos or images containing the same object are not available. Moreover, objects have much less articulation and pose changes as compared to humans undergoing complex actions. In contrast, the method we proposed in this dissertation performs colocalization in action videos which are much more complex than videos containing moving objects. Furthermore, we also propose a method where localization can be achieved using a single video.

2.3.3 Co-segmentation

A related problem to localization includes video object co-segmentation. As compared to object segmentation in a single video, co-segmentation employed multiple videos to obtain pixel location of the same common object in multiple videos. Most of the current co-segmentation methods have underlying assumption of one dominant object in each video. The authors in [99] obtain an object proposal in each frame and track them forward and backward over the video. The final segmentation is achieved using shape, color and motion similarities in a regulated maximum weight clique's framework. Similarly, authors in [18] produced accurate co-segmentation of a moving object in a video using shape and color similarities employing conditional random field (CRF).

There are several critical differences between cosegmentation approaches and the methods proposed in this dissertations. Co-segmentation approaches compute object proposals in each frame and track them through video using motion and location similarities. In contrast, we employ action proposals which are produced through hierarchical clustering with explicit camera motion compensation and are much more robust against abrupt camera motion and complex articulations of humans [27, 49, 81]. Second, while matching objects across videos, co-segmentation approaches find similar tracklets across videos. Instead, we match spatio-temporal volumes using global fea-

tures, which are more robust as compared to matching simple tracklets which are susceptible to error due to the complex articulation of humans. Third, similar color matching; a strong similarity cue for co-segmentation; cannot be used for matching actions across videos as the similar actions are usually performed by people wearing different color clothes. Fourth, we employ fine-grained matching which matches motion patterns of different parts of humans. Furthermore, both of the above co-segmentation methods have been tested on clean videos while we test the proposed approach on challenging action datasets containing low-quality videos that have large variations in pose, cluttered backgrounds, and poor illumination conditions.

2.4 Leveraging images and videos jointly

Although images and videos belong to different domains, they may contain complementary information. For example, videos contain objects under different articulations and poses while images capture key poses, canonical viewpoints, and important temporal instances of an action or event. Therefore, both images and videos can be used jointly to obtain improved results either in one domain or in both. For instance, videos can be used to increase the training data for an object detector by providing object's examples in several different poses. Similarly, images can help to overcome the noise in video data by providing key instances of the action and hence help in training robust detector from videos. In our work, we leverage the discriminative information available in the images to obtain action localization in videos.

Below we describe the recent works which leverage videos for images and images for videos respectively.

2.4.1 Leveraging Videos for images

There is a large amount of weakly labeled data available in YouTube. This weakly labeled video data can be used to augment image data for improved object and action classification. Furthermore, as compared to images, videos have temporal consistency information which can be exploited to remove noise and obtain good training examples. Prest et al. [51] obtained YouTube videos of similar content and localize similar looking objects jointly using multiple videos. They use appearance and motion similarities to localize similar looking objects. Since image and videos belong to the different domain, they perform domain adaptation on videos. Finally, they demonstrated that the object classifiers trained on both image and video data perform better than the classifiers that are trained only on images. Chen $et\ al.$ [5] used unlabeled videos to learn action detectors for images. They augment their training data by discovering video frames that are similar to images. After finding similar looking video frames, they also use their temporally neighboring frames (forward and backward) to get more data. They show improved image action classification results by levering this video data. Recently, the authors in [88] demonstrated an unsupervised learning of deep network from image classification by exploiting the video data in an unsupervised manner. They obtained thousands of YouTube videos and track small patches in each video. By tracking the patches, they gather millions of similar looking patches. Finally, they used those similar looking patches in triplet neural network to learn feature representation. After learning feature representations from videos, they fine-tuned over the target dataset and showed excellent classification results. *However, we are not aware of any previous work that uses images to localize an action in a video.*

2.4.2 Leveraging Images for Videos

Images have been used recently for solving several problems in the video domain. Several new algorithms has been introduced which utilize images to perform action or event recognition [6, 28, 74] and video summarizations [33, 34, 65]. Tang et al. [74] devised a method to adapt object detector from images to videos. They iteratively adapt object classifier trained on images by repeatedly discovering similar examples from un-labeled videos using self-paced learning scheme. They demonstrated excellent results for detecting objects in videos using object detector trained originally from images. Authors in [6] proposed to use weakly labeled web videos and web images to perform complex event recognition in consumer videos. They designed multi-domain adaptation with heterogeneous sources to learn optimal classifiers to classify similar concepts in unlabeled target data. Jain et al. [28] provide usefulness of using 15,000 object categories for action detection. They demonstrate that using objects related to action not only provide compact representation but also improve classification results. Recently Sun et al. [71] performed fine-grained temporal action localization in long untrimmed videos by employing domain transfer from web images. They used images to localize action specific frames in videos and then used those localized frames to train action recognition models using long short-term memory networks.

Recently, several methods have been proposed to obtain video summarization using images. Khosla et al. [33] presented an approach where images are used as prior data to video summarization. They exploit the fact that images are taken to capture the most informative instance of the events and hence can be used to extract key information from the video. Song et al. [65] reported a title based video summarization approach. They used images and video frames jointly to remove redundant and non-informative frames from the video to produce effective video summarization. Kim et al. [34] presented an efficient framework to produce a joint summary of the videos and Flicker images. They used Flicker images to obtain high-quality video summarization and exploit video to

obtain storyline graphs of images. In contrast to above-mentioned papers, we have solved a more complex problem of action localization using web images.

2.5 Summay

In this chapter, we have reviewed recent important approaches introduced for action recognition. We first reviewed some recent approaches for action recognition and localization and its improvements such as foreground focused action recognition. Next, we introduced recent several weakly supervised and col-localization approaches. Moreover, we briefly described methods that used videos and images jointly. We also highlighted key differences between related works and methods proposed in this dissertation. In the next chapter, we describe our approach for foreground-focused action recognition along with its cross-dataset recognition applications in details.

CHAPTER 3: ACTION RECOGNITION ACROSS DATASETS BY FOREGROUND-WEIGHTED HISTOGRAM DECOMPOSITION

In general, the action classifiers learned on foreground action regions are expected to have less dataset bias and hence can generalize well across different datasets. Also, background is sometimes useful to discriminate between similar action classes. However, such background objects or scene information should either be visually salient, in motion or comprised of objects that can be detected explicitly. Otherwise, it is difficult to differentiate between useful background and features that are independent of action being performed. The background should be diverse as well, but *not* discriminative, i.e., it should not aid in recognition of the action class, otherwise it would limit the generalizability of the class model, and consequently result in worse cross-dataset recognition than within dataset. One of our key objectives, in this chapter, is to demonstrate that background/scene information artificially inflates accuracy within datasets and inhibits meaningful generic representation of actions. Although the recent datasets are much larger and complex as compared to simple initial action datasets, backgrounds of some datasets are much more discriminative i.e., only background itself is discriminate enough to classify an action video even without knowing anything about the action. In fact, recent works [85, 87] have demonstrated that dense features (computed at exhaustively different scales and locations) significantly outperform interest point based methods [11, 90, 43]. However, the dense action classifier trained on one dataset does not necessarily perform that well when tested on the different action dataset for the same action.

Based on insights from the analysis in the first part of this chapter, we propose a scheme for weighting local features that does not require actor detection and tracking for solving action recognition across datasets problem. We propose to use low-level unsupervised visual cues to obtain the probability of each pixel being representing foreground. We first use these weights in codebook

formation so that foreground features contribute more during clustering. In our weighted histogram representation, different words contribute to histogram differently based on their probability of being foreground. Moreover, we use these foreground weights during the computation of similarities between videos. Specifically, we divide videos into different regions based on the probability of being foreground and the similarity between two videos is the weighted summation of similarities between regions that corresponds to the same quantized probability of being foreground.

3.1 Background Discriminativity in Action Datasets

Action recognition dataset should be representative of our surrounding visual world, and therefore as diverse as possible. Besides illumination, clutter, etc., the sample actions should capture variations due to viewpoint, pose, speed, and articulation. The datasets with discriminative backgrounds scenes are good choice for video classification and scene classification but not for action classification. This is because the datasets with discriminative backgrounds scenes will inhibit the development of true action representations. In this section, we quantify the discriminative power of background scenes in a few well-known action datasets using two methods. First, we compute motion features for only background regions to perform recognition and second, we measure class-wise confusion between action classes using global scene descriptor.

3.1.1 Background Motion Features

Computation of background features in a video requires annotation or estimation of regions corresponding to the action. For this analysis, two recent datasets are selected: UCF Sports [54] and UCF YouTube [44], due to the availability of manually annotated actor bounding boxes.

Dense space-time interest point descriptor (STIP) [43] are extracted for all videos in both datasets.

Table 3.1: Accuracy using STIP in different video regions. There is little decrease in performance even when completely ignoring features on the actor. In UCF Sports, background only features actually perform better than foreground only features.

STIP Sampling	**UCF Sports**	**UCF YouTube**
Foreground only	71.92%	59.80%
Background only	73.97%	55.27%
Foreground and Background	75.34%	60.60%

The features are extracted with a 50% spatio-temporal overlap, using a single spatial and temporal scale. The features with an overlap of more than 50% with the actor bounding boxes are then labeled as foreground, while all remaining features are considered background features. Note that multi-scale features would not allow such categorization. We choose STIP because of their good performance. The train/test process follow the original papers, i.e., leave one actor out classification for UCF Sports, and leave one group out classification for UCF YouTube. We use Bag-of-Words representation to represent each video. Bag-of-Words representation was the state-of-the-art representation for different computer vision applications before deep learning. The experimental results for both datasets are shown in Table 3.1. It is evident that even complete removal of foreground words does not have a significant detrimental effect on accuracy. It is reasonable to assume that the action in an arbitrary video can hypothetically be replaced with a different action without a significant change in background feature descriptors. The implication, then, is that the background alone is almost as discriminative as the action itself. An action model trained on these datasets with dense feature coverage will, therefore, perform poorly on a novel test set with a different background composition or distribution.

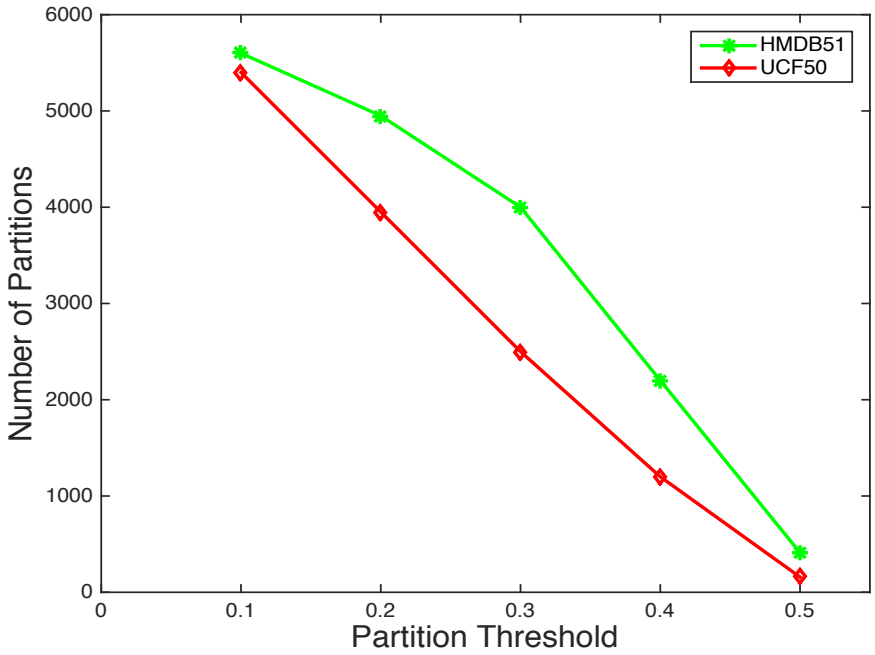

Figure 3.1: Relative number of partitions c of frames in UCF50, and HMDB51 datasets as a function of inter-partition distances, τ. At the maximum allowed distance (0.5), UCF50, and HMDB51 have 158, and 411 partitions respectively. Please see text for interpretation.

3.1.2 Global Scene Features

Using a global image descriptor to represent an action video has two inherent disadvantages. First, it would ignore the motion or temporal properties of the video, and secondly, it may not appropriately capture the background scene since a significant part of every image can potentially be the actor. If a scene descriptor, despite these limitations, can be used to reasonably recognize an action, the corresponding classifier is likely to generalize poorly.

We use the GIST descriptor [48] to quantify discriminativity of background scenes in action datasets. The GIST descriptor in this case has the advantage that annotation is not required to represent the background. The GIST descriptor is computed for every 50^{th} frame of all videos in

UCF50 [52], and HMDB51 [37] datasets.

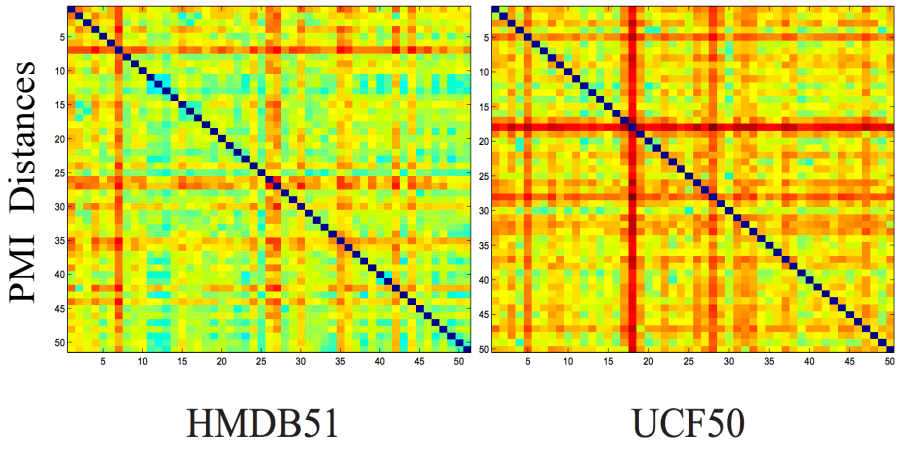

Figure 3.2: Class-wise PMI distance matrices for each dataset can be considered as the inverse of a confusion matrix. The values are normalized with respect to the maximum of across two datasets, so the same colors correspond to the same absolute value.

The first experiment we perform is to quantify the relative number of distinct background scenes in each dataset at a fixed level of separation in the feature space. Given n GIST descriptors in a dataset, a graph $G = (V, E)$ is constructed, such that $V = \{v_i\}$, $i \in \{1, \ldots, n\}$, is the set of all descriptors, and $E = \{e_{ij}\}$, $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, n\}$, and $e_{ij} = \|v_i - v_j\|_2$ is the Euclidean distance between descriptors i and j. The feature distance matrix, E is then thresholded to obtain U such that $u_{ij} = 1$ for all *i* and *j* , if $e_{ij} \leq \tau$, and 0 otherwise. The graph connected component analysis of U then results in a partitioning of the n GIST descriptors into c groups. Obviously, the number of partitions c approaches 1 as threshold τ increases, and it approaches n as τ decreases.

The number of partitions obtained in this manner is independent of n, and for a specific τ provides a comparison of two datasets in terms of the relative diversity of scenes. For any dataset, the closest points in any arbitrary pair of partitions within the dataset have a distance of at least τ. Therefore, the larger the value of c, the more diverse the backgrounds will be. A quantitative comparison of the number of partitions in each of UCF50, and HMDB51 datasets is shown in

Fig. 3.1. One would expect that UCF50 and HMDB51 with the similarly large number of classes, and complexity should have the similar number of partitions in the GIST feature space, at equal inter-partition distances. However, HMDB51 has consistently larger values of c for the same τ, as compared to UCF50. This comparison points to the hypothesis that the background scene features would be less helpful for the former dataset. A similar observation was made in [37] when using scene descriptors.

The number of partitions c does not explicitly reveal the relative importance of background in discriminating action classes. Therefore, we perform another experiment where we cluster the GIST descriptors for each dataset into k clusters using k-means. We then compute the pointwise mutual information (PMI) between each cluster and an action class, resulting in a $k \times N$ matrix, where N is the number of classes in the dataset. Each k-long column in the matrix is a representation of the corresponding action class in terms of the k clusters, and can be used to compare it with other classes. We then compute the class-wise Euclidean distances between all pairs of classes to obtain an $N \times N$ matrix, P (see Fig. 3.2 for examples). Each element p_{ij} of the matrix P represents the discrimination between GIST-based representations of classes i and j. The larger the value, the easier it is for GIST to classify a test action video. We compute the mean discrimination as the average of matrix P. These values for different values of k are reported in Table 3.2. The relative confusions for HMDB51 and UCF50 are more similar, with the former being consistently harder than the latter.

Table 3.2: Average discrimination between classes of different datasets, using PMI of GIST clusters, for different number of clusters. Discrimination increases (confusion decreases) as number of clusters (background scene codebook size) increases. UCF50 and HMDB51 are comparable, the latter being consistently harder.

k	**100**	**200**	**300**
HMDB51	7.15	11.07	14.06
UCF50	7.97	11.79	14.38

3.2 Foreground specific Action Representation

Given our experimental verification of the effect of scene background on action classification, we propose to learn foreground specific action representation to improve recognition on novel test sets. However, the problems of foreground-background segmentation and human or actor detection are very challenging, and all the more so, in unconstrained videos that make up the more recent action datasets. Since our eventual goal is to recognize actions, rather than segmentation, or actor detection, our proposed framework does not attempt to label each pixel or region as foreground or background. Instead, we estimate the confidence in each pixel being a part of the foreground and use it directly to obtain the codebook as well as the video representation. This confidence is computed using several cues as explained below.

3.2.1 Motion Gradients

Action is mainly characterized by the motion of moving parts. We use this important cue to assign high confidence to the locations undergoing articulated motion in a video. Under the fixed camera assumption, we can simply use magnitude of optical flow to estimate the probable location of the actor. However, since most of the realistic datasets involve moving camera, simple optical flow magnitude can be high for background as well. Hence, we use the Frobenius norm of optical flow gradients. The motion gradients based foreground confidence, f_m is then defined as:

$$\begin{aligned} f_m(x,y) &= \left\| \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix} \right\|_F * g \\ &= \sqrt{u_x^2 + u_y^2 + v_x^2 + v_y^2} * g, \end{aligned} \tag{3.1}$$

where, u_x, v_x, u_y, and v_y are the horizontal and vertical gradients of optical flow respectively, and g is a 2D Gaussian filter with a fixed variance. The main idea behind using optical flow gradients is that it not only helps to remove camera motion but also gives high magnitude around the articulated moving object. A qualitative example of $f_m(x, y)$ for a frame in a moving camera video is shown in Fig. 3.3.

3.2.2 Color Gradients

In many videos, the actor has different appearance and color than the background, while the background (such as sky or floor) has relatively uniform color distribution. Therefore, the color gradients can be used as a cue towards estimating the confidence in the location of actor and object boundaries, while resulting in low responses for background regions with uniform colors. Specifically, we compute the color gradient based confidence in observing a foreground pixel, f_c, using the Frobenius norm of LAB color space given as:

$$f_c(x, y) = \sqrt{L_x^2 + L_y^2 + a_x^2 + a_y^2 + b_x^2 + b_y^2} \ * \ g, \tag{3.2}$$

where (L_x, a_x, b_x) is the horizontal gradient of the color vector at (x, y). A qualitative example of $f_c(x, y)$ is shown in Fig. 3.3.

3.2.3 Saliency

We use visual saliency as the third cue to estimate the confidence in observing a foreground pixel. Visual saliency attempts to replicate the human visual system in selecting regions of interest in complex scenes. In sports videos (a common type of actions in UCF50, HMDB51 and Olympic sports), the player receives the most of the visual attention and hence represents the most salient part of the video. A similar observation applies to amateur as well as professional moving camera

videos that follow objects with distinct appearance amid relatively homogeneous backgrounds. Although our ultimate goal is to estimate foreground confidences for *videos*, we observed that, due to large camera motion and noisy optical flow, video or motion based saliency methods do not always result in reasonable outputs. To address such difficulties, we use graph-based visual saliency [24] to capture the salient regions in each frame individually. We chose this method due its computational efficiency, evident capability in finding salient regions and natural interpretation as decomposition of the image into the neural network.

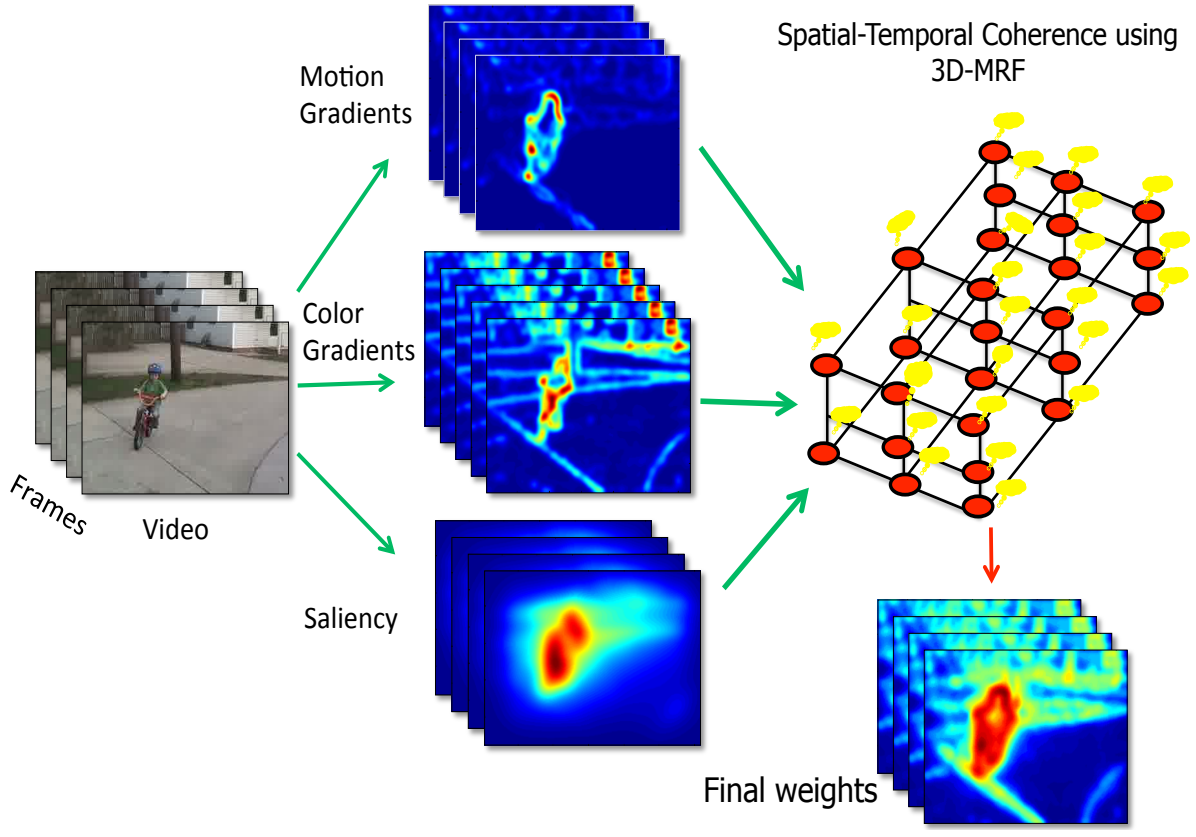

Figure 3.3: This figure shows original video frame on the left; optical flow gradient magnitude $f_m(x, y)$; magnitude of color gradient in LAB space $f_c(x, y)$; and saliency $f_s(x, y)$ in middle and final weights after 3D-MRF on right.

Following [24], we compute contrast, luminance, and four orientation maps corresponding to ori-

entation $\theta = \{0^o, 45^o, 90^o, 135^o\}$ using Gabor filters, all on multiple spatial scales. In the activation step, a fully connected directed graph is built where edge weight between two nodes, corresponding to pixel locations, (i, j) and (p, q) is given as

$$B_a(i,j,p,q) = \Big|M(i,j) - M(p,q)\Big| \; exp\left(-\frac{(i-p)^2 + (j-q)^2}{2\varphi^2}\right), \tag{3.3}$$

where $M(i, j)$ represents the features at (i, j), and φ is a free parameter. Using the graph we define a Markov chain, then the stationary distribution of the chain is computed and treated as an activation map, $A(p, q)$. A new graph is then defined on all pixels with the edge weights being:

$$B_n(i,j,p,q) = A(p,q)\; exp\left(-\frac{(i-p)^2 + (j-q)^2}{2\varphi^2}\right). \tag{3.4}$$

Again, the weights of outbound edges are normalized and the graph is treated as a Markov chain. The equilibrium distribution of the chain is then used as per pixel saliency measure, $f_s(x, y)$. An example of the final saliency based foreground confidence is shown in Fig. 3.3. In that, high values (red color) represents the salient region in each frame.

3.2.4 Coherence of Foreground Confidence

Since saliency and color gradients are computed based on a single frame, they ignore the temporal information as well as coherency. Moreover, color as well as optical flow computation does not explicitly impose spatial coherence constraint. We therefore compute an initial aggregate confidence map, $\hat{f}_a$, as: $log\left(f_m\left(f_c + f_s\right) + 1\right)$. The values of $\hat{f}_a$ are max-normalized for each frame of a video. In order to impose spatio-temporal dependency among neighboring pixels, we use tem-

poral extension of 2D Markov random field [15] similar to [95]. The video is considered as a 3D grid graph, $(\mathcal{V}, \mathcal{E})$, where each node is connected to four spatial and two temporal neighbors. If a labeling ω assigns a weight $\omega_p \in \Omega = [0, 1]$ to a node, $\psi_p \in \mathcal{V}$, then the quality of labeling is given by the following energy function:

$$E(\omega) = \sum_{\psi_p \in \mathcal{V}} D_p(\omega_p) + \sum_{(p,q) \in \mathcal{V}} V(\omega_p - \omega_q). \tag{3.5}$$

We use quadratic data term defined as $D_p(\omega_p) = (\hat{f}_a(p) - \omega_p)^2$ and truncated quadratic smoothness term given as $V(\omega_p - \omega_q) = min((\omega_p - \omega_q)^2, \kappa)$. For inference, we use well know message passing algorithm. During inference, in addition to spatial neighbors, each node receives a message from the temporal neighbors as well. At time t, the message, $m^t_{p \to q}(\omega_q)$, that node p sends to q is given as

$$\min_{\omega_p} \left(D_p(\omega_p) + V(\omega_p - \omega_q) + \sum_{s \in \mathcal{N}_p \backslash q} m^{t-1}_{s \to p}(\omega_p) \right), \tag{3.6}$$

where $\mathcal{N}_p \backslash q$ represents the five neighbors of p other than q and the belief vector for node q at time t is,

$$b^t_q(\omega_q) = D_q(\omega_q) + \sum_{s \in N_q} m^t_{s \to q}(\omega_q). \tag{3.7}$$

The inference starts by sweeping in six directions, and after a fix number of iterations, the weights which give the minimum cost for the belief vector are selected as final, and denoted as f_a. Qualitative examples of the final confidence after 3D MRF is shown in Fig. 3.4. We summarize foreground estimation technique in Figure 3.3. Note that after imposition of spatio-temporal dependencies the resultant map assigns high weights for the entire region corresponding to the actor and the bicycle and gives low values for the background pixels despite the camera motion.

Figure 3.4: Original video frame, and corresponding confidence of each pixel being the foreground in eight example videos from UCF101 and JHMDB datasets are shown.

3.3 Foreground weighted Representation

Given the confidence in each pixel being in a foreground region, we modify the bag-of-words representation of a video in several important ways. Our goal is to represent the video so that features corresponding to the actual action, i.e., the foreground, contribute towards the vocabulary

as well as the resulting representation, while those in the background have minimal effect when training models or comparing videos. Our proposed ideas are described in the following.

Weighted Codebook: Traditionally, the codebook or vocabulary in a bag-of-words framework is learned using *k*-means, whereby the set of feature descriptors $X = \{\mathrm{x}_i\}$, obtained from training videos are the data points to be clustered into k clusters, with centers represented as z_j. Given the pixel-wise foreground confidence for every frame in every video, we begin by computing the average foreground confidence, $w_i = \sum_{(x,y)\in P_i} f_a(x,y)/|P_i|$, where P_i is the set of pixels in the spatiotemporal volume corresponding to descriptor x_i. We then employ weighted K-means to obtain the codebook, where the goal of clustering is to minimize the following energy function,

$$\underset{C}{argmin} \sum_{j=1}^{K} C(i,j) w_i \left\| \mathrm{x}_i - \mathrm{z}_j \right\|^2, \tag{3.8}$$

where C is the $|X| \times k$ unknown membership matrix. The resulting vocabulary, $Z = \{\mathrm{z}_j\}$, is a set of points in the feature space that are more similar to descriptors with high confidence of being a foreground, and potentially farther away from descriptors on the background.

Weighted Histogram: In the bag-of-words method, the image or video is represented as a k long vector, $H = [h^1, \ldots, h^K]$, where h^j is the number of times the nearest neighbor of a descriptor in the video is found to be z_j. Given the average foreground confidence, w_i for the descriptor, x_i, we propose to compute the weighted histogram, $\hat{H}$, where $\hat{h}^j$ is the sum of w_i for all descriptors whose nearest codeword is z_j. The weighted histogram, therefore, is influenced by features with high confidence of being in the foreground regions, while the background features have a minimal effect on the final video representation. The weighted histogram is not to be confused with soft quantization where the weight w_i would have been distributed across bins.

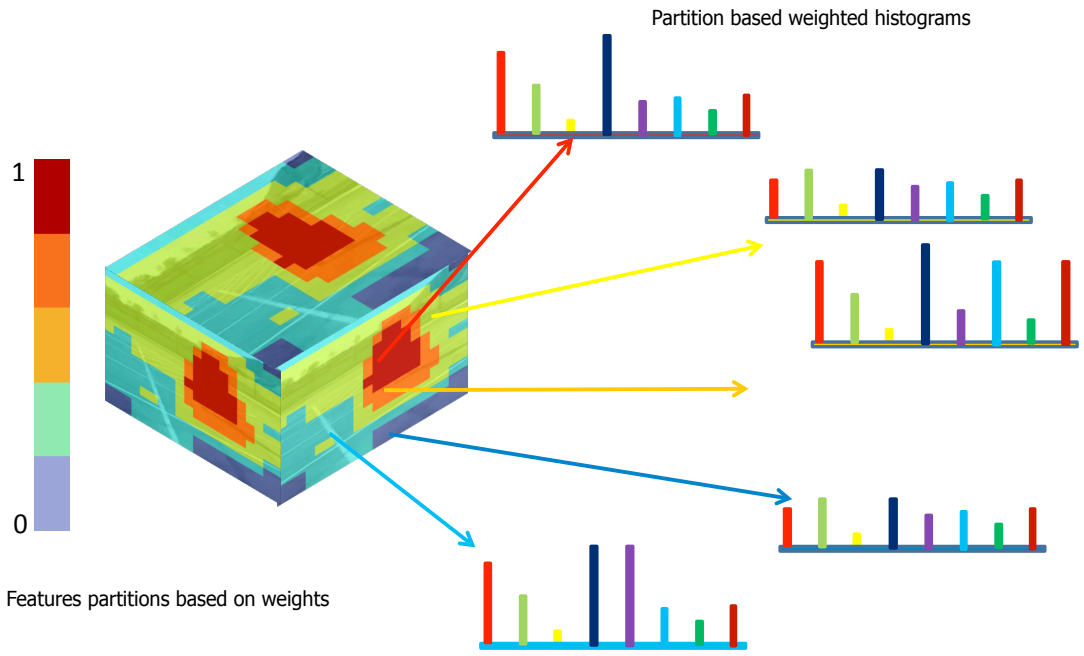

Figure 3.5: Illustration of foreground confidence based histogram decomposition.

3.3.1 Foreground Confidence based Histogram Decomposition:

We notice that despite weighing the influence of features on the histogram, the accumulative effect of background features on different bins of the histogram can sum up to be significant. This is because a significant number of pixels in the video, and consequently densely samples descriptors, can have relatively low foreground confidence. In other words, the number of high confidence features contributing to the histogram is far less than those with low confidence of being foreground. This is not a problem if features with high and low confidences are quantized to different words, but that may not always be the case, especially due to the weighted codebook.

If the foreground and background regions were divided into two distinct classes (binary labeled), it would be straightforward to compute two different histograms for each type of region. However, to avoid thresholding and binarization of foreground confidence, we propose a novel alternative

solution. We begin by categorizing the spatiotemporal regions corresponding to different feature descriptors into R classes. These classes correspond to R equal, non-overlapping, exhaustive partitions of the range of average foreground confidences, w. A set $\hat{\mathbf{H}}$ of R weighted histograms, $\hat{H}_r$, is then computed for all the features in each of the R groups separately. The following kernel function then replaces histogram intersection:

$$\Delta\left(\hat{\mathbf{H}}^i, \hat{\mathbf{H}}^j\right) = \sum_{r=1}^{R} \alpha_r \Theta\left(\hat{H}_r^i, \hat{H}_r^j\right), \tag{3.9}$$

where Θ is the histogram intersection kernel, and α_r are predefined weights, that increase linearly with $r \in \{1, \ldots, R\}$. As a result, regions of two videos that have approximately the same foreground confidence, are compared only with each other. This process and the effect of our proposed scheme are illustrated in Fig. 3.6. As can be seen in the figure, the proposed multiple weighted histograms and the weighted average of histogram intersection kernel, show obvious improvement as a representation and measure of similarity between videos, respectively. The illustration of foreground histogram decomposition is shown in figure 3.5.

3.4 Experiments

The objective of our experiments is to verify that models trained using features from foreground regions are likely to generalize better and attain higher recognition accuracy than dense sampling, especially when tested on videos from novel, unseen datasets. To this end, we performe extensive experiments, evaluating the effect of our proposed foreground confidence measure in a weighted bag-of-words framework for cross-dataset recognition over three datasets: UCF50, HMDB51, and Olympic sports.

UCF50 has 50 action categories. Since all the videos in UCF50 are taken from Youtube, they are

implicitly biased towards a specific type of video shooting, including but not limited to amateur shooting style, cluttered background, and abrupt camera motion. Videos in 51 action categories of HMDB51 are mostly taken from movies and a small number from YouTube and Google Videos. The two datasets have 10 actions with common class labels, namely basketball, biking, pull-ups, golf swing, horse riding, punch, fencing, push ups, rock climbing, and walking.

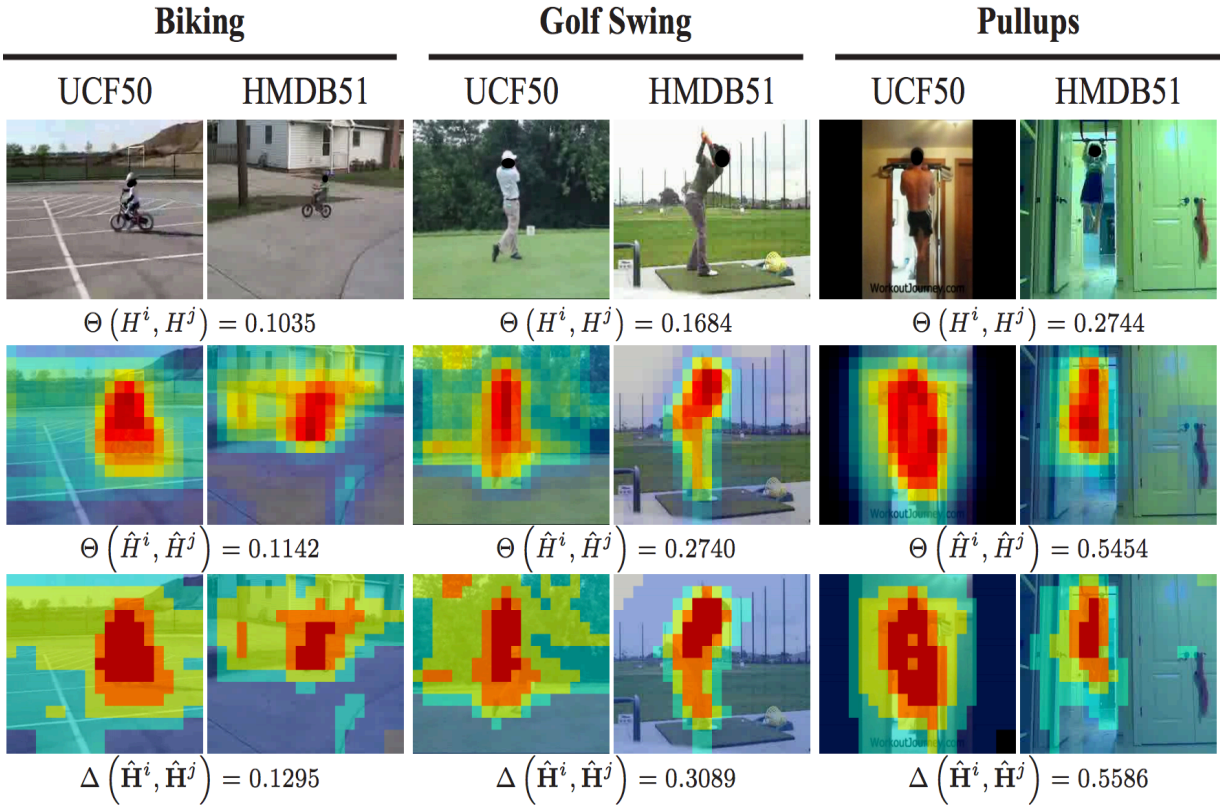

Figure 3.6: Illustrate of the the effect of weighted histograms, and foreground confidence based decomposed histograms. In each column, the top row shows the original image. The middle row shows the relative average foreground confidence or weight w_i of each spatiotemporal cuboid in the video, where shades of red correspond to high values. The third row shows the category or group, $r \in \{1, \ldots, R\}$, out of a total of $R = 5$. Features in each group are compared only with those in corresponding group in other videos. Notice that similarity between same label videos increases with use of weighted histograms, $\hat{H}$, and the proposed kernel Δ for decomposed histograms, $\hat{\mathbf{H}}$.

In our experiments, we only chose the first 5 classes which are *visually* similar in both datasets.

Table 3.3: Average accuracy of action recognition across different pairs of training and testing datasets. 'Unweighted' is the traditional bag-of-words paradigm, using dense STIP features. The column labeled 'Weighted' corresponds to foreground confidence weighted vocabulary and weighted histograms. The column labeled 'Histogram Decomposition' uses multiple histograms for different range of foreground confidence values, and uses a weighted mean of individual histogram intersections as the kernel. As can be observed, our two proposed representations perform significantly better than the baseline for most experiments.

Training	**Testing**	**Unweighted (%)**	**Weighted(%)**	**Histogram(%) Decomposition**
UCF50	UCF50	70.00	74.20	77.85
UCF50	HMDB51	55.70	60.00	68.70
HMDB51	HMDB51	65.30	69.30	68.00
HMDB51	UCF50	66.43	67.86	71.43
Olympic Sports	Olympic Sports	71.80	73.95	69.79
UCF50	Olympic Sports	30.2	32.29	34.37
Olympic Sports	UCF50	16.67	28.12	52.08

We do not consider other actions because even though they may have similar class labels, they are visually very different: almost all the videos of punch in UCF50 correspond to the sport of boxing in a boxing ring, while most of the HMDB51 punch videos are more unconstrained, such as those from fist fights. Similarly, the walking action of HMDB51 is quite different from the 'walking with dog' action of UCF50. We only want to evaluate videos from the same class which is at least visually/semantically similar to a human observer. Specifically, for experiments between UCF50 and HMDB51, we use biking, golf swing, pull ups, horse riding, and basketball. We use basketball, pole vault, tennis serve, diving, clean & jerk and throw discus for experiments between Olympic sports and UCF50 datasets.

For training and testing on HMDB51, we use the train and test partitions for each of the chosen action as the original setup [37]. In order to have a fair comparison, we select the same number

of training and testing videos from UCF50 by dividing 100 videos of each class into 18 and 7 groups respectively, i.e., an $\sim$70-30 ratio. We compute dense STIP descriptors over both datasets with cuboid size of 32 by 32 spatially, and 15 frames temporally, with 50% spatial and temporal overlap.

For all datasets, we train multi-class classifiers. A 3000-words vocabulary is created using K-means algorithm. As in traditional bag-of-words approach, all the STIP features are quantized into 3000-bin histograms for each video, to establish the baseline representation, H. For weighted histograms $\hat{H}$, instead of having an equal contribution, each descriptor contributed to the histogram based on its confidence in being the foreground. Finally, for foreground based histogram decomposition $\hat{\mathbf{H}}$, we divide the set of all features in each video into $R = 5$ groups based on foreground weights. We use $\alpha_i = \{1, 0.8, 0.6, 0.4, 0.2\}$, for weighted summation of the 5 histogram intersections.

The *Olympic dataset* consists of 16 human actions, where the videos mostly depict athletes practicing different sports actions. This dataset is also collected from Youtube. Similar to our previous experiment, we use 6 common actions between UCF50 and Olympic Sports (see Table 3.3 for labels). Due to the small number of training and testing examples in Olympic Sports (40 training, 10 testing examples), we extend the dataset by adding a horizontally flipped version of each video sequence. In order to ensure fairness, we use the same number of videos from UCF50 and add their horizontally flipped counterparts. We ensure that both the original and flipped video pairs are in either training or testing sets but not both.

Our experimental results are reported in Figure 3.7, Figure 3.6, and Table 3.3. It can be seen from the confusion matrices in Figure 3.7 corresponding to the performance of UCF50-trained classifiers on HMDB51 test videos, compared to the traditional bag-of-words, our proposed representations exhibit consistent improvement in all action classes except basketball. When using baseline clas-

sifiers trained on UCF50, 68% of the horse riding examples in HMDB51 are classified as biking. Using the proposed method, however, we are able to reduce this confusion to 36%. In the complementary experiment, baseline classifiers trained on HMDB51 categorized 57% of UCF50 biking examples as horse riding, while the proposed method reduced the confusion to 30%. Similarly, reducing dependency on the background has significantly improved accuracy for pull ups and golf swing, where the actions are visually similar across the datasets. The drop in basketball accuracy is likely due to variation in actor pose and viewpoint across datasets.

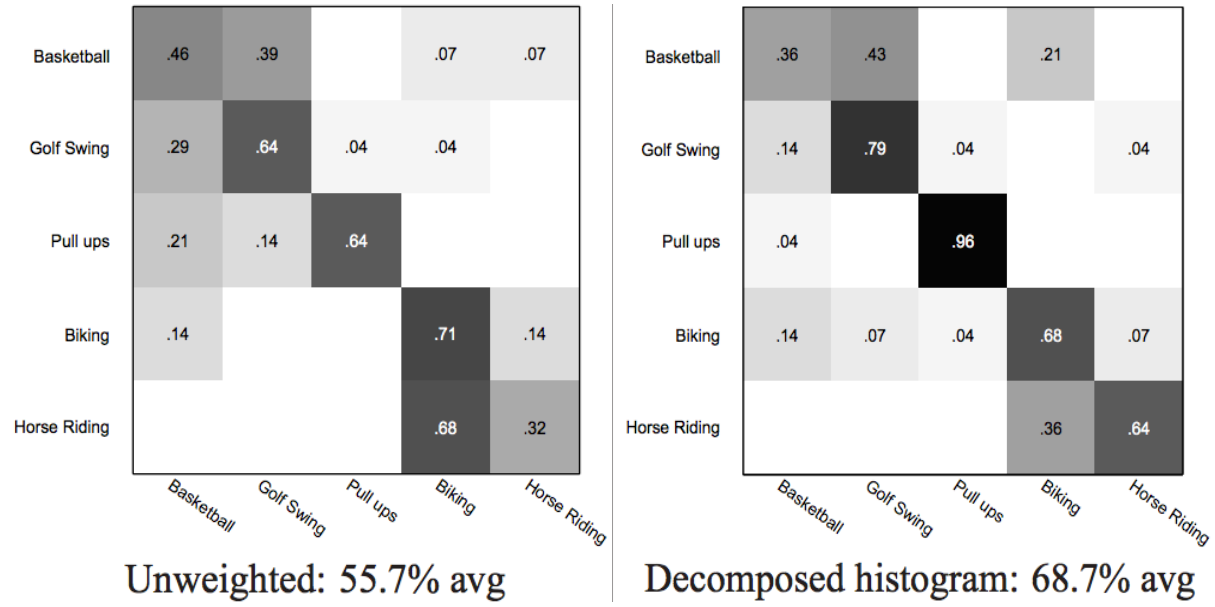

Figure 3.7: confusion tables for unweighted and decomposed weighted histogram classifiers trained on UCF50 and tested on HMDB51.

As reported in Table 3.3, the quantitative results conclude that the proposed framework for estimation of foreground confidence is meaningful.The consistently higher recognition accuracies serve as an empirical verification of our conjecture that the dataset-specific background scenes are one of the main causes of deterioration in recognition accuracy across datasets. Moreover, when training and testing on distinct datasets, the histogram decomposition, and the newly proposed corresponding similarity measure perform better than even the foreground weighted vocabulary and

histograms, for all cross-dataset experiments.

3.5 Summary

We attempt cross-dataset action recognition without using labels or features from the test set. In doing so, we experimentally demonstrate the detrimental effect of background scenes on action recognition dataset. We also proposed a new process for obtaining per pixel confidence of every video pixel being the foreground, as well as novel soft assignment, and histogram decomposition schemes for the bag-of-words representation. Our extensive experimental results and discussion validate the proposed ideas and framework.

In addition to facilitating cross-dataset action recognition, estimating foreground probability of every pixel has several applications such as in action localization and action annotations. In the next chapter, we exploit these foreground confidences to obtain action annotations in realistic action videos.

CHAPTER 4: AUTOMATIC ACTION ANNOTATION IN WEAKLY LABELED VIDEOS

Training a robust action detection system requires hundreds or thousands of spatio-temporal action annotations. These annotations require hundreds of human hours and expensive annotations interfaces. With the recent explosive growth of action datasets such as THUMOS14 dataset [31], it is prohibitive to manually obtain spatio-temporal bounding box annotations for each video. Furthermore, the resurgence of deep learning algorithms, which need thousands or even millions of annotations to achieve good detection accuracy enforce the needs of automatic annotation systems. In this chapter, we exploit foreground confidences, introduced in the last chapter, to obtain automatic action annotations.

We propose a weakly labeled approach to obtain spatio-temporal annotations of videos given video level labels only. In the proposed approach, we begin with obtaining action proposals in a video. In each video, we rank action proposals using elementary action cues and select a few high-quality action proposals from several thousand proposals. Then, for multiple videos of the same action, we compute the similarity between proposals across videos by carefully considering their saliency, shape, and fine grained similarity. Finally, by using the similarity information among multiple proposals in a global framework, we select the most action representative proposals in each video. Figure 4.1 provide the pictorial summary of our approach.

4.1 Action Proposals

The first step of our approach is to generate spatio-temporal action proposals. To achieve this, we obtained action proposals using improved dense trajectories using an unsupervised method

[81]. First, we employ unsupervised hierarchical clustering algorithm [61] to merge dense trajectories using HOG, HOF, MBH, Traj and SPAT (spatio-temporal positions) features. Then, to achieve efficiency and spatio-temporal smoothness, only a few nearest neighbors are considered to compute similarity, while merging clusters. Finally, the clusters whose distance is more than a certain threshold represents individual action proposals. As compared to previous methods [49, 27], this method [81] does not require supervoxel segmentation and has time and space complexity of $O(n^2)$. Figure 4.2 shows typical action proposals for UCF-Sports dataset.

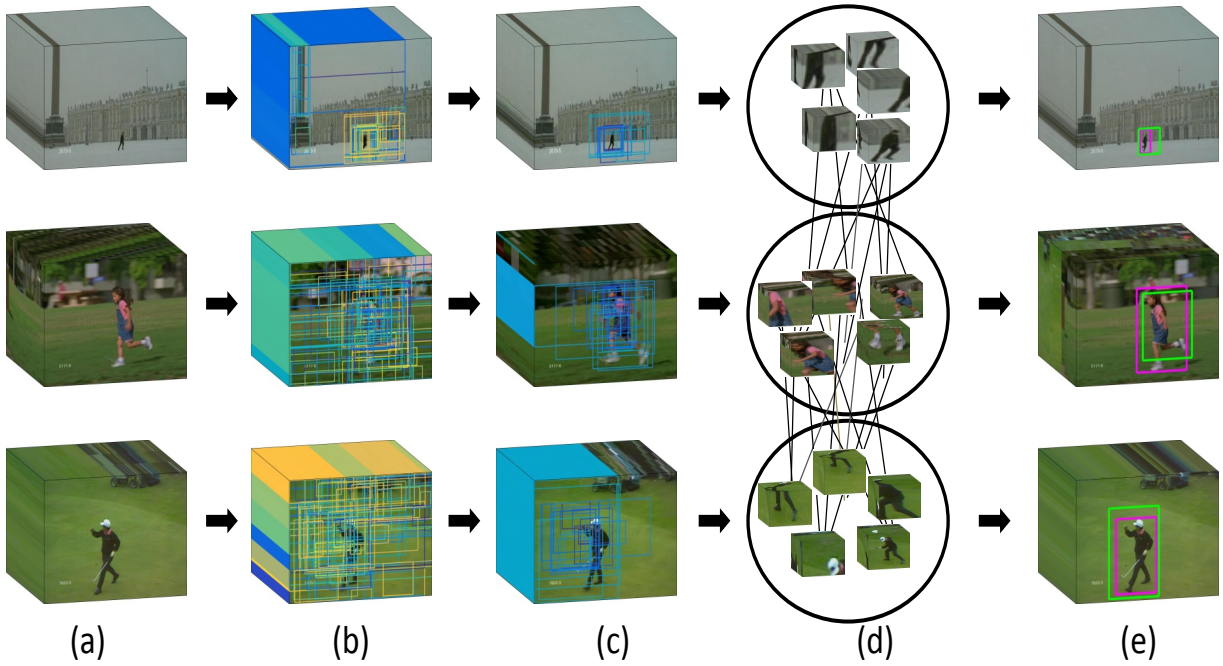

Figure 4.1: Block diagram of our approach. (a) Given the multiple videos of the same action ('running' in this figure), (b) We first compute large number of action proposals in each video (section 4.1) (c) After that we obtain a few most action representative proposals in each video using motion and saliency information employing MAP based proposal subset process (section 4.1.1), (d) Then, we construct a fully connected graph between proposals across multiple videos, where edge between proposals captures global, fine grained and shape similarities between proposals (section 4.2), (e) Finally, using generalized maximum clique of this graph , we obtain the most action representative proposal in each video (section 4.3). Colors of proposals are randomly selected except (e) where magenta shows ground truth and green box represents automatically discovered action proposal.

We stress that our main approach of action annotation does not depend on any specific action proposal method and any recent action proposal methods [27, 49, 81] can be employed.

4.1.1 Initial Proposals Ranking

Although action proposals reduce search space for action detection and classification, the numbers of proposals are still huge and cannot be directly used in place of action annotation. However, given a large number of proposals, one can safely expect at least a few proposals that would have very high overlap with the actual actor spatio-temporal location. Our ultimate goal is to discover those action representative proposals automatically.

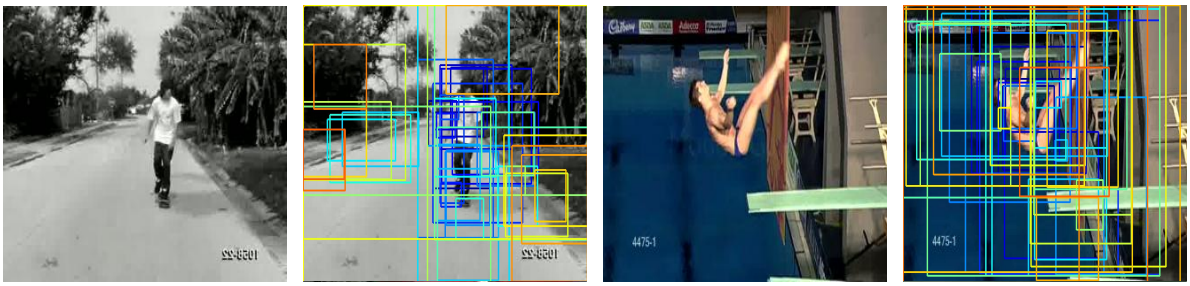

Figure 4.2: Typical action proposals. Color of proposals is randomly assigned.

Not all action proposals are equally important. Many proposals originate from the background and several contain only part of an action. Moreover, computing expensive features from all proposals is computationally inefficient. Therefore, we propose to rank action proposals using simple elementary features and keep a few highly action representative proposals, only. Inspired by [69, 1], we compute the following action cues from each video independently.

Motion Cues: Motion boundaries have proven to be resistant to camera and background motion but characterize human motion quite well [86]. Therefore, we compute Frobenius norm of optical

flow to estimate the probable location of an actor.

$$\|U_I\|_F = \sqrt{u_x^2 + u_y^2 + v_x^2 + v_y^2} \tag{4.1}$$

where U_I represents forward optical flow of frame *I*, $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$ and u_x, v_x, u_y and v_y are their optical flow gradients along x and y axis respectively.

Visual saliency: Actors usually stand out among their neighbor and capture visual attention. We estimate saliency of each pixel in the video frame using [24]. In this method, feature and orientation maps are computed at multiple scales using local gradients and Gabor filters, respectively. Finally, center surround activation maps and their normalization are obtained using a fully connected graph over feature space. Further details of this method can be found in [24].

Spatio-temporal coherence: The above features are estimated independently for each frame and max normalized to represent foreground score maps. We aggregate motion (M) and saliency (S) as: M+S. These initial scores have no or little spatial temporal coherence. Therefore, we impose spatio-temporal consistency using a discontinuity preserving 3D Markov Random Field framework, which enforces smoothness in nearby video locations. The video is considered as a 3D grid graph, where each node (pixel) is connected to four spatial neighbors and two temporal neighbors. . Formally, 3D MRF energy minimization is given as,

$$E(l) = \sum_{p \in \mathcal{V}} \Phi(l_p) + \sum_{(p,q) \in \mathcal{N}} \Psi(l_p - l_q), \tag{4.2}$$

where l_p is labelling (score) of pixel *p*. We use quadratic unary term Φ and truncated quadratic smoothness term Ψ. The inference over this graph is achieved using Max-Product/Min-Sum loop belief propagation [16, 69]. Qualitative examples of foreground scores, in Figure 4.3, for moving a camera and low-quality videos demonstrate the robustness of the above framework for estimating foreground regions.

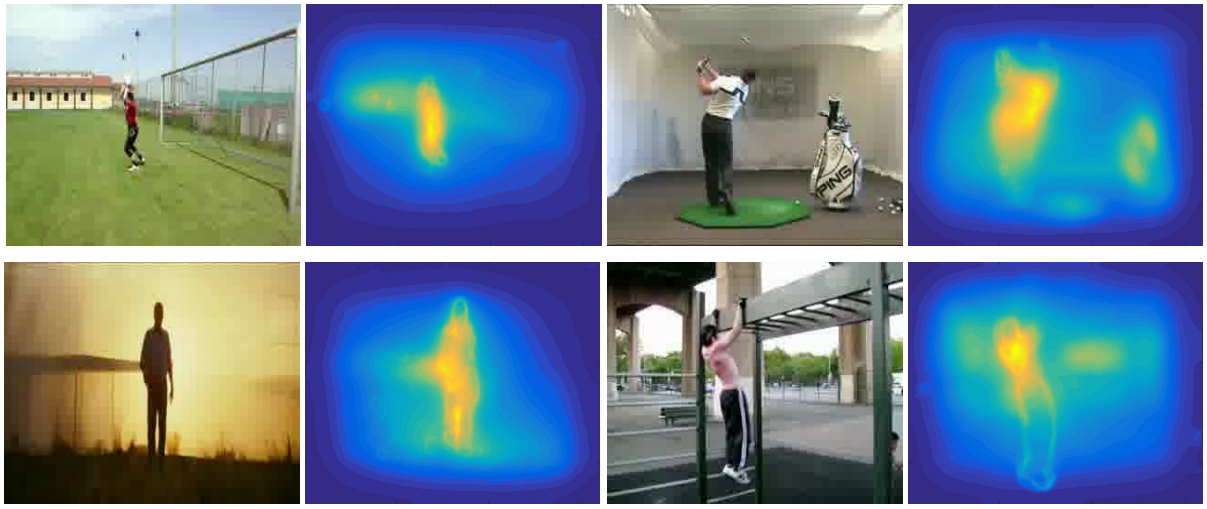

Figure 4.3: Action Score Map for four different actions videos of sub-JHMDB dataset.

Finally, we estimate initial action score, Ω_{p_i}, of each proposal by computing foreground score (normalized by proposal area) within each proposal.

Given all proposals $\mathcal{P}$ in a video $\mathcal{V}$, we use this initial action score to select a few most probable action proposals. However, this initial action score of the proposal can be noisy and also there may be many highly overlapped proposals, therefore to select a small subset $\mathcal{S}$ of most probable action proposals, we propose to use MAP-based proposal subset selection procedure similar to [100].

Overall, we want to group proposals into automatically determined number of clusters and select exemplar proposal from each cluster. We assume an auxiliary variable $\mathbf{Z}$= $(z_i)_i^n$, where z_i =j if proposal p_i belongs to a cluster represented by proposal p_j and n corresponds to the total number of action proposals in the video. The joint distribution of a few selected proposals $\mathcal{S}$ and $\mathbf{Z}$ for video $\mathcal{V}$ is given by:

$$P(\mathcal{S}, \mathbf{Z}|\mathcal{V}) = \frac{P(\mathcal{V}|\mathcal{S}, \mathbf{Z})P(\mathcal{S}, \mathbf{Z})}{P(\mathcal{V})}, \tag{4.3}$$

where $P(\mathcal{V}|\mathcal{S},\mathbf{Z})$ represents likelihood term and $P(\mathcal{S},\mathbf{Z})$ represents prior term. $\mathcal{S}$ is binary vector of dimension n and contains 0 and 1s. $\mathcal{S}_i$=1 means proposal p_i is selected. We want to estimate Maximum a posteriori (MAP) of above equation. In order to select a few action representative less overlapping proposals, the prior term can be written as

$$P(\mathcal{S},\mathbf{Z}) = K_1 P(\mathbf{Z}))\mathcal{W}(\mathcal{S})\mathcal{C}(\mathcal{S},\mathbf{Z}), \tag{4.4}$$

where K_1 makes the prior term a valid probability mass function. The constraint indicator vector $\mathcal{C}(\mathcal{S},\mathbf{Z})$ is 1 for exemplar proposals p_j, which means there exist at least z_i such $z_i = j$. Note that $\mathcal{C}(\mathcal{S},\mathbf{Z})$ is 0 for non-exemplar proposals. $\mathcal{W}(\mathcal{S})$ is prior information about detection window and is given as:

$$\mathcal{W} = W_1 \times W_2, \tag{4.5}$$

W_1 softly penalizes highly overlapped proposal windows in $\mathcal{S}$ and is given by:

$$W_1 = \prod_{i,j:i\neq j} exp(-\gamma \times IOU(p_i, p_j)), \tag{4.6}$$

where $IOU(p_i, p_j)$ represents *intersection over union* between two proposals. $W_2 = exp(-\phi N)$ controls the number of finally selected proposals. We choose this parameter so that at least one hundred proposals in each video are produced. After substituting prior and likelihood term, Equation 3 can be written as:

$$P(\mathcal{S},\mathbf{Z}|\mathcal{V}) \propto P(\mathbf{Z}|\mathcal{V})\mathcal{W}(\mathcal{S})\mathcal{C}(\mathcal{S},\mathbf{Z}), \tag{4.7}$$

where $\mathcal{W}(\mathcal{S})$ and $\mathcal{C}(\mathcal{S},\mathbf{Z})$ are defined above and considering the independent assumption among z_i, $P(\mathbf{Z}|\mathcal{V})$ can be given as follows:

$$P(\mathbf{Z}|\mathcal{V}) = \prod_{i=1}^{n} P(z_i|\mathcal{V}). \tag{4.8}$$

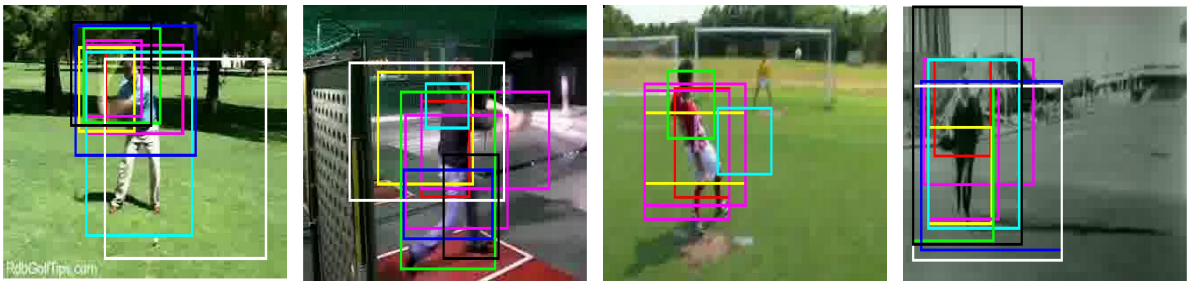

Figure 4.4: Top few action proposals for four actions videos of sub-JHMDB dataset.

In the above equation, $P(z_i = j|\mathcal{V}) = \lambda \times K_2$ if $j = 0$, otherwise it is $K_2 \times IOU(p_i, p_j) \times \Omega_{p_i}$, where K_2 is a normalization constant and Ω_{p_i} is the initial score of proposal p_i. Note that $P(z_i = j|\mathcal{V})$ encourages proposals, which have high overlap with many initial highly action scored proposals and hence it is robust to noisy scored proposals. Figure 4.4, shows a few top ranked action proposals after proposal subset selection process.

4.2 Proposals Similarity Across Multiple Videos

Although a few top ranked proposals maintain high Mean Average Best Overlap (MABO, defined in Equation 4.12), the top most proposal does not necessarily represent the best available proposal. Therefore, we re-rank the proposals by leveraging action proposals similarity across multiple videos of the same action.

A naive similarity measure between proposals can hurt the proposal ranking, since; sport videos backgrounds are more similar than the action itself. Therefore, we use global, fine grained and shape similarities between proposals to discover the most action representative proposals.

Each of the similarity measures between proposals is explained below.

Global Similarity

We use the bag of words (BOW) similarity between proposals. We represent each proposal by M-bin global histogram and spatial pyramids of 2×2 using improved dense trajectory features (Trajectory, MBH, HOF, and HOG) [86]. Next, the similarity between two proposals is measured using χ^2- distance, which is defined as:

$$S_{ij}^{f} = exp\left(-\gamma \sum_{k=1}^{k=d} \frac{(h_{ik} - h_{jk})^2}{(h_{ik} + h_{jk})}\right), \tag{4.9}$$

where h_i and h_j respectively, represent bag of words histogram of feature f for i^{th} and j^{th} proposal and d is the dimensions of the histogram. The final similarity between any two proposals, $\mathbf{\Theta_{ij}}$, is the linear combination of individual feature similarities.

Fine Grained Similarity

Proposal matching using spatial pyramid (the fixed grid structure) has an underlying assumption that similar action parts appear at the same location in both proposals. However, due to actor articulations, large camera motion, pose and scale variations, the fixed location assumption is not always true. Therefore, we propose the use of flexible matching between action regions to obtain aggregated similarity between action regions as a proposals similarity measure. Since the flexible similarity measure takes into account the similarity across the local region between proposals, we call it fine grained similarity measure.

To achieve this, we cluster *raw* improved dense trajectory features within each proposal in C_r clusters, where subscript *r* corresponds to MBH, HOG, HOF or Trajectory. Note that we cluster all four features (MBH, HOG, HOF, and Trajectory) separately. A $C_r \times C_r$ distance matrix is computed using Euclidean distance. To allow the flexibility in matching between different spatio-temporal action regions, we use only raw features (without their actual coordinates) during Euclidean dis-

tance computation. Finally, the optimal one-to-one matching between clusters across two proposals is obtained using Hungarian algorithm [38]. We compute the similarity between clusters of each raw feature separately and final similarity, $\mathbf{\Gamma_{ij}}$, is a linear combination of all of them. Figure 4.5 illustrates the flexible matching of clusters (the same color) across proposals. We have experimented with different values of C_r but found our results insensitive to the exact value of C_r. In experiments, we arbitrarily choose C_r = 6.

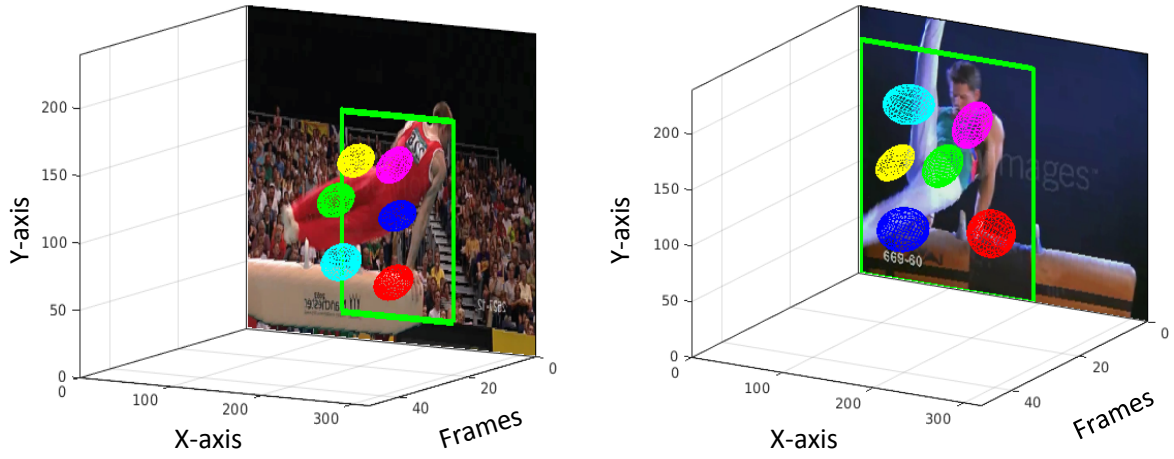

Figure 4.5: Illustration of fine-grained matching across videos. The figure shows the matching of motion patterns produced by hands and torso (red and magenta ellipses), abdomen (green and yellow ellipses) and legs (blue and cyan ellipses) of two actors in different videos.

Proposal Shape Similarity

In addition to spatio-temporal features within action proposals, the shape of proposal windows (height, width and aspect ratio) over time itself carries useful information about an action. Mostly the same action in multiple videos undergo through similar articulations and therefore, the similar shape proposal windows across videos likely capture the same action.

We define the shape of action proposal p_x and p_y over time as:

$$\begin{aligned} \Lambda_{p_x} &= [r_1, r_2, \ldots, r_n], \\ \Lambda_{p_y} &= [r_1, r_2, \ldots, r_m], \end{aligned} \tag{4.10}$$

where $r_i = \frac{w_i}{h_i}$ and w_i and h_i are the width and height of proposal in frame i and m, n are length of proposals. A naive way to match shape of p_x and p_y is to match r values frame by frame. However, the same action can occur with different speeds in different videos and therefore, in most cases $n \neq m$. Hence, we propose to consider proposals' shape Λ_{p_x} and Λ_{p_y} as time series and find similarity, $\mathbf{\Pi_{ij}}$, between them using dynamic time warping.

4.3 Generalized Maximum Clique Graph Optimization

Given each proposal action score, as well as their pairwise similarity across multiple videos of the same action, we seek to identify the most action representative proposals from every video that have a high action score as well as a high similarity with highly action representative proposals in other videos. Due to large intra-class variation, matching only two videos may not necessarily facilitate better localization. Therefore, we re-rank all the videos jointly using a global framework. To this end, a fully connected graph **Z = (V, E)** is constructed, such that $\mathbf{V} = \{v_i\}$, $i \in \{1, \ldots, n\}$, is the set of all proposals, and $\mathbf{E} = \{e_{ij}\}$, $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, n\}$, represents the edge

between p_i and p_j, where $e_{ij} = \eta_{pi} \times \eta_{pj}(\mathbf{\Theta_{ij}} + \mathbf{\Gamma_{ij}} + \mathbf{\Pi_{ij}})$. η_{pi} discourage proposals whose length is very small as compared to the length of video and is given by: $exp(-(m-n)/n)$, where m is length of video and n is length of proposal.

We divide all nodes (which correspond to proposals) into disjoint groups, where each group Z_i belongs to one action video. The nodes within each group is a set of all top ranked proposals in a single video. We call them a group because they belong to the same video. Since we want to select one node from each group, the feasible solution is a subgraph that satisfies two constraints: 1) Only one node from each group is selected; 2) If one node is included in feasible solution, then its $N-1$ edges to single node in each of $N-1$ groups should be included as well.

Formally, the feasible solution can be found by maximizing the following objective function:

$$\sum_{i=1}^{i=N} \sum_{j=1, j\neq i}^{j=N} \left(\alpha \Omega_{p_i} + (\mathbf{\Theta_{ij}} + \mathbf{\Gamma_{ij}} + \mathbf{\Pi_{ij}}) \eta_{\mathbf{pi}} \times \eta_{\mathbf{pj}} \right), \tag{4.11}$$

where Ω_{p_i} represents the initial action score of i^{th} proposal, $\mathbf{\Theta_{ij}}$, $\mathbf{\Gamma_{ij}}$ and $\mathbf{\Pi_{ij}}$ show the similarity between proposals computed in previous section, η_{pi} discourage small length proposals and α controls the weight of the initial action score for the final objective function.

The optimal solution is a subgraph that maximizes the above objective function. It is easy to observe that the above combinatorial optimization problem falls under the umbrella of generalized maximum clique problem (GMCP) [98][17]. GMCP is the class of graph theory problems that generalizes the standard subgraph problems (from node to group of nodes). The input to GMCP is graph $\mathbf{Z}$, as defined above. Specifically, the input graph consists of groups of nodes where edges exist between all the proposals across the groups only with no connection within the group. The output of GMCP is a subgraph $\mathbf{Y}_s = (V_s, E_s)$, such that each node in the subgraph belongs to one video only and the objective function is maximized.

GMCP is an NP-hard problem. We use the approximate solver proposed in [98], for which code is available online. This local neighborhood solver has fast convergence speed and is memory efficient. Specifically, we initialize the initial solution from the top ranked proposals (from section 3.2) and generated $N_Z \times Z_N$ local feasible solutions of size 1, where N_Z denotes the total number of groups (number of videos of an action) and Z_N represents the number of nodes in a group (number of proposals in a single video). The solution that has the maximum score is selected and again $N_Z \times Z_N$ local feasible solutions are generated around this newly found solution and so on. We repeat this process until we reach the maximum number of iterations or no more updated solution can be obtained with further iterations.

The above formulation of GMCP assumes only one action instance in a video. However, there are some videos in UCF-Sports and THUMOS'13 datasets, which have multiple instances of an action in the same video. To annotate multiple action instances in a video, we use GMCP iteratively. During each iteration (after the first one), we stop the node selection for the videos that have all of its instances annotated. For the videos which have yet more instances to be annotated, we ignore the nodes that have high overlap with an already selected nodes (as they may be localizing the same instance) and find GMCP solution from rest of the nodes. We repeat this process until all instances in all videos are annotated.

4.4 Experimental Results

We evaluate our method on three action datasets: UCF-Sports [54], sub-JHMDB [37, 30] and THUMOS13 [25]. These datasets are among the most challenging action datasets. Ground truth bounding box annotations are available for all three datasets. Note that for all three datasets, we re-size frames to 240×320.

In all experiments, we compute action proposals using online implementation of [81]. Improved dense trajectory features are extracted using [86] and encoded in the standard bag of words paradigm. The value of α in Equation 4.11, which controls the contribution of initial ranking, is set to 0.07.

UCF-Sports [54] contains 10 human actions. This dataset includes actions such as diving, kicking, lifting, horse riding, etc., and has 15 videos per action. These low-quality YouTube videos contain huge camera motion, dynamic backgrounds, viewpoint changes and large intra-class variations.

sub-JHMDB [30] contains 12 complex human actions. This dataset includes actions such as catch, climb stairs, run, jump, swing basketball etc. This dataset is a subset of JHMDB [30] and contains 316 videos. On average, there are 26 videos in each action class. As mentioned in [30], this subset is far more challenging as compared to the whole JHMDB dataset.

THUMOS13 [25] is the largest and the most challenging trimmed action detection dataset with 24 complex human actions. It includes actions such as pole vault, skiing, ski-jet, surfing, fencing, cricket bowling etc. This dataset is a subset of UCF-101 and includes 3207 videos with multiple instances of an action in the same video. This dataset contains 133 videos for each action class.

In the next three subsections, first, we evaluate initial proposal ranking followed by qualitative and quantitative analysis of localization results and finally present their detailed analysis.

4.4.1 Evaluation of Initial Proposal Ranking

Following previous works [51, 27, 79], we evaluate robustness of our initial action proposal ranking using Mean Average Best Overlap (MABO). MABO measures the quality of the best available proposal. To compute MABO, we first compute mean of best overlap (ABO) for each action class

c as follows:

$$ABO_c = \frac{1}{|K_c|} \sum_{g_i^c \in G_c} \max_{p_j \in P} O(g_i^c, p_j), \tag{4.12}$$

where g_i^c represents ground truth annotation for i^{th} video in class c and p_j is the j^{th} action proposal from **P** proposals in a video. $|K_c|$ is the total number of ground truth in class c. The overlap O is computed using standard *intersection over union* for each frame and averaged by the number of frames where either g_i^c or p_i exist. Finally, MABO is mean of ABO over all action classes. First row in Table 4.1 shows the MABO calculated using only top 100 proposals. The second row presents MABO calculated using all action proposals for all three datasets. It is impressive to note in Table 4.1 that even by using 10% proposals, sufficiently high MABO is obtained. This indicates that we have at least one good quality proposal among top-ranked proposals.

Note that although initial proposal ranking maintains high MABO for top 100 proposals, the top most proposal has significantly low MABO (UCF-Sports: 18.54, sub-JHMDB: 31.25, THUMOS'13: 21.01). Therefore, to achieve better localization, we can perform matching among top ranked proposals only and can ignore the processing of several thousand proposals.

Table 4.1: First two rows illustrate MABO of top ranked proposals using Non Maximal Suppression (NMS) and the proposed approach respectively. The bottom row shows the MABO using all proposals in a video. On average, UCF-Sports, sub-JHMDB and THUMOS13, respectively, contain 1866, 328 and 2300 proposals in every video.

Proposals	**UCF-Sports**	**sub-JHMDB**	**THUMOS13**
Non Maximal Suppression (NMS)	51.98	50.74	33.60
Ours	56.01	55.25	35.26
All (Upper bound)	62.40	57.77	46.71

4.4.2 Localization Results

In this section, we describe our experimental results for weakly labeled action localization using multiple videos. Note that in weakly labeled action *localization*, we want to discover spatio-temporal location of the action in the video for which we already know that the specific action is happening in that video. At start, each video contains on average 1866, 328, and 2300 action proposals in UCF Sports, sub-JHMDB and THUMOS13, respectively. After initial ranking, we select the top 100 proposals from each video from UCF Sports and top 50 proposals (to reduce computation) from sub-JHMDB and THUMOS13.

Following several previous action localization methods [42, 76, 27], we use intersection-over-union criterion at an overlap of 20% for correct action localization. The quantitative results for three challenging action datasets are shown in Table 4.2 (last row).
The numbers in the table show the percentage of the videos that have correct localization. Note that we obtain these localization results without any training videos.

The qualitative localization results for all action classes of UCF Sports sub-JHMDB are shown in Figures 4.6, 4.7 and 4.8, 4.9 respectively. In these figures, magenta bounding box represents ground truth annotations and green bounding box shows automatic annotation.

Table 4.2: Localization results and comparison with related work. The numbers inside brackets show MABO (defined in equation 4.12).

Method	**UCF-Sports**	**sub-JHMDB**	**THUMOS13**
Cosegmentation [99]	53.15 (23.3)	49.37 (24.5)	9.56 (5.48)
Cosegmentation [18]	42.25 (19.2)	47.78 (20.3)	21.41 (12.4)
Siva et al. [63]	48.49 (21.8)	61.39 (22.3)	14.39 (10.2)
Tang et al. [75]	61.18 (27.9)	64.56 (22.8)	14.17 (10.0)
Ours	**83.83 (34.6)**	**89.56 (32.4)**	**41.69 (19.1)**

Figure 4.6: Qualitative results of five action of UCF-Sports. Every three frames show an action video at different time instances. Green and magenta bounding boxes show proposed automatic annotation and manual annotation, respectively.

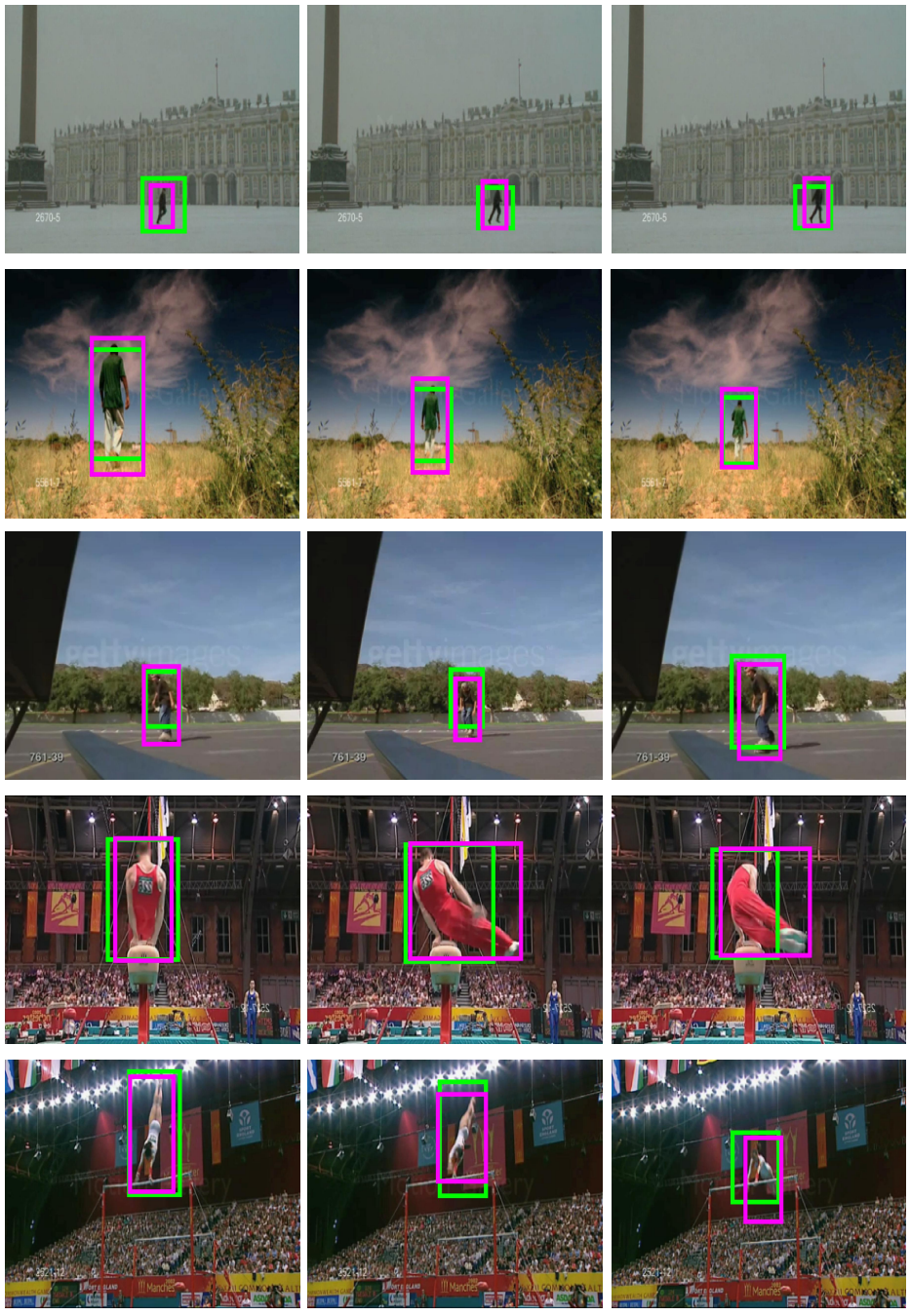

Figure 4.7: Qualitative results of five action of UCF-Sports. Each row show three frames of an action video at different time instances. Green and magenta bounding boxes show proposed automatic annotation and manual annotation, respectively.

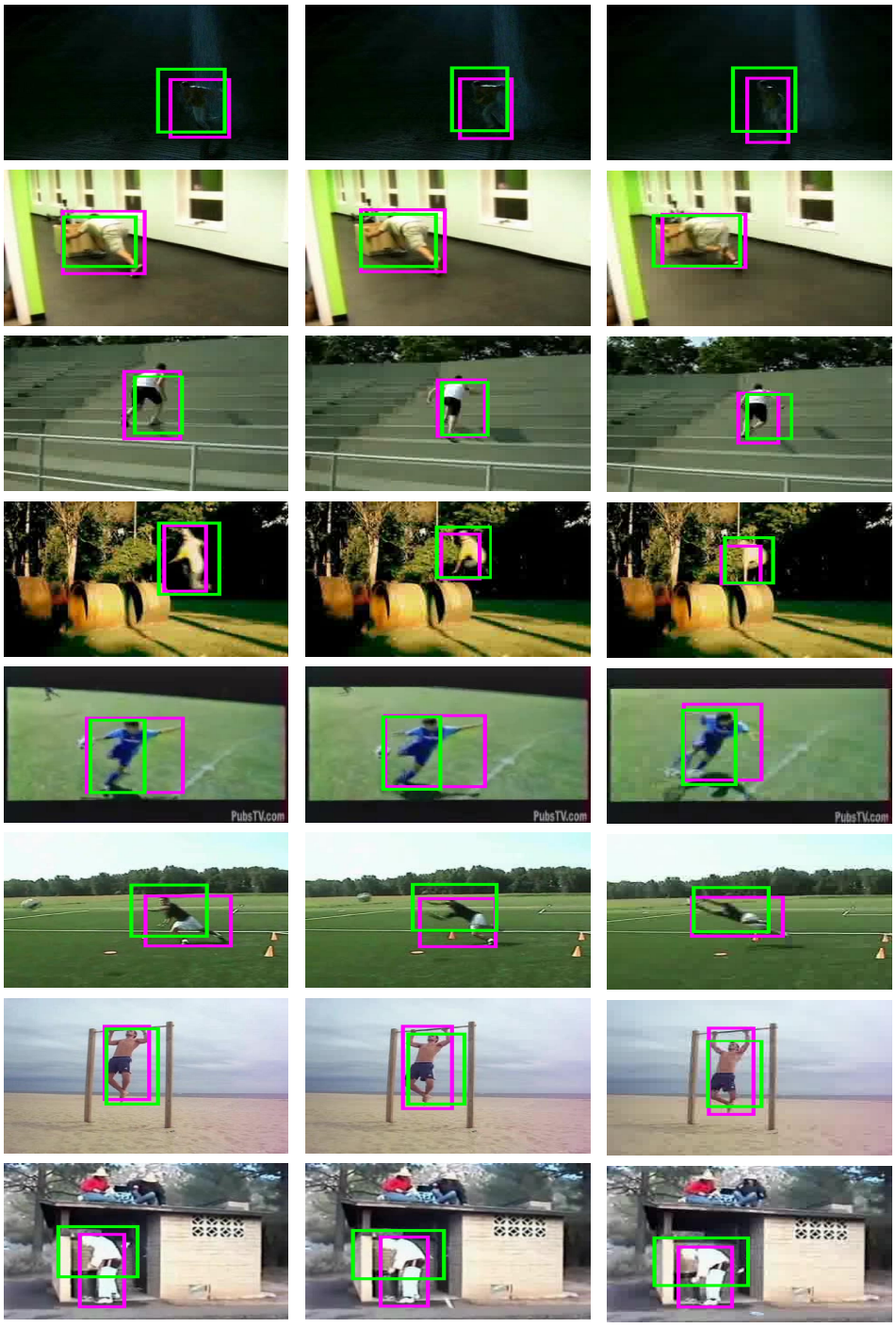

Figure 4.8: Qualitative results for all actions of sub-JHMDB. Each row show three frames of an action video at different time instances. Green and magenta bounding boxes show proposed automatic annotation and manual annotation, respectively.

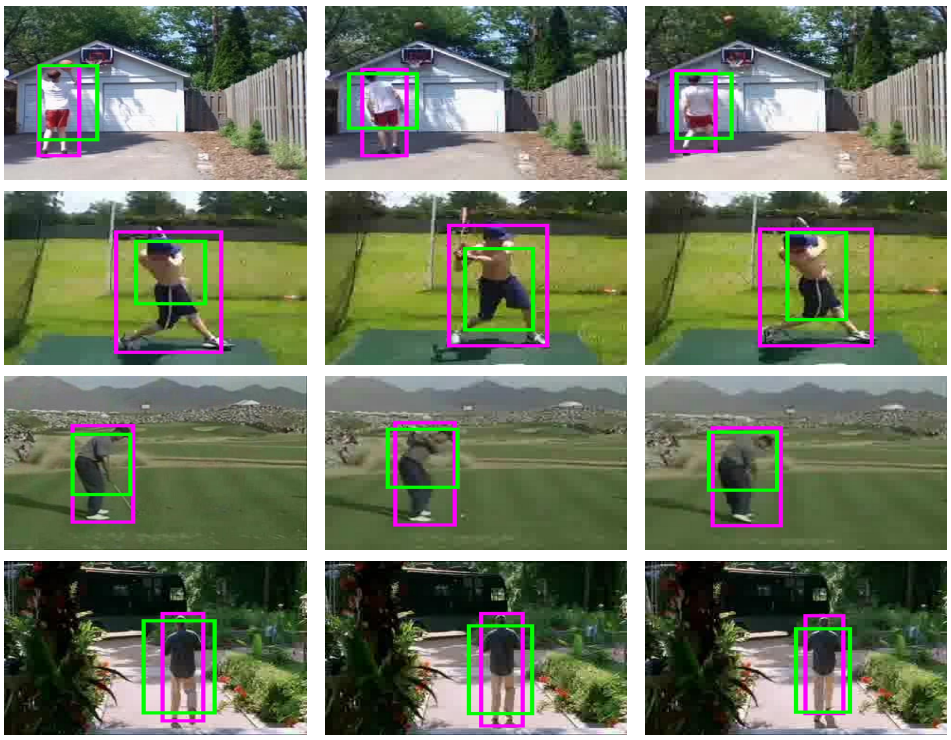

Figure 4.9: Qualitative results for all actions of sub-JHMDB. Each row show three frames of an action video at different time instances. Green and magenta bounding boxes show proposed automatic annotation and manual annotation, respectively.

Note that despite large camera motion and change in scale, fast and abrupt motion, background clutter occlusion, our automatic annotations closely follow the manual annotations.

4.4.3 Comparison with related works

Our approach is closely related to video object co-segmentation. For this purpose, we use recently published video object co-segmentation methods [18, 99]. Using the codes available on the authors website, we produced the co-segmentation for each video and put a tight bounding box around the segmented region to represent co-localization. Experimental results of these methods are given in

4.2.

Our work is also related to the weakly supervised concept or action detection using negative mining [75, 63]. Similar to ours, both methods assume the availability of video level labels. In addition, they use negative data to localize the main concept or action in the video. We follow the procedure described in [75] and use videos not labeled with the action of interest as negative data. We use both methods to discover the best representative proposals among top-ranked proposals. Experimental results of [75, 63] on all three datasets are given in Table 4.2. The quantitative comparisons in Table 4.2 demonstrate the superiority of our approach. The localization accuracy of our approach at various localization threshold is given in Table 4.3

Table 4.3: Localization accuracy of UCF-Sports, sub-JHMDB and THUMOS13 at various localization thresholds.

Threshold	0.2	0.3	0.4	0.5
UCF-Sports	83.83	63.26	33.76	16.74
Sub-JHMDB	89.56	61.08	21.52	4.75
THUMOS13	41.69	23.07	10.11	3.90

Since the most important entity in the action videos is the humans; it would be interesting to see how stat-of-art human detector performs as compared to our approach. For this purpose, we use recent Faster-rcnn human detector [53] and detect the human in each frame of the video and make the proposal. The quantitative results in Table 4.4 demonstrate the superiority of ours approach. Note that human detector [53] is trained on thousands of precisely annotated examples using advanced deep learning tools. Moreover, training of human detector needs very high computation power i.e., GPUs. In contrast, we just use video level labels and handcrafted features and our approach do not need GPUs.

Table 4.4: Comparision with Human Detector (Faster R-CNN) [53]

Method	**UCF Sports**	**sub-JHMDB**	**THUMOS13**
Human Detector [53]	76.88	87.66	35.86
Ours	**85.29**	**90.51**	**41.69**

4.4.4 Action Detection

In this section, we demonstrate the usefulness of automatically obtained bounding boxes by using them to train action detectors. We use fisher vector representation and apply PCA on the improved dense trajectory features to reduce dimensions to half. After that, we randomly sample features to learn Gaussian Mixture Model (GMM) with 128 Gaussians. Finally, we apply power and l_2 normalization to fisher vector of each proposal. We use one-versus-all linear SVM to learn action classifier by considering automatically obtained annotations from action class as positive examples. For the negative example, we take annotated proposals from negative classes. Note that the automatic annotation is done using training videos only; *no test video* is used during annotation. For all three datasets, we use training-testing partitions as recommended by the authors of datasets. We train separate classifiers using exactly the same settings for annotations obtained by two baseline methods [75, 63]. We evaluate detection accuracy using ROC (False positive rate versus True positive rate), AUC (Area under ROC curves), PR (Precision Recall) as well as mAP (mean Average Precision) metrics. Following several previous works [27, 42, 76], we use overlap of 20% for correct localization. Experimental results, in Figure. 4.10, on three datasets using standard evaluation metrics, demonstrate that our approach significantly outperforms the baselines on sub-JHMDB and THUMOS13 and have comparable results on UCF-Sports dataset.

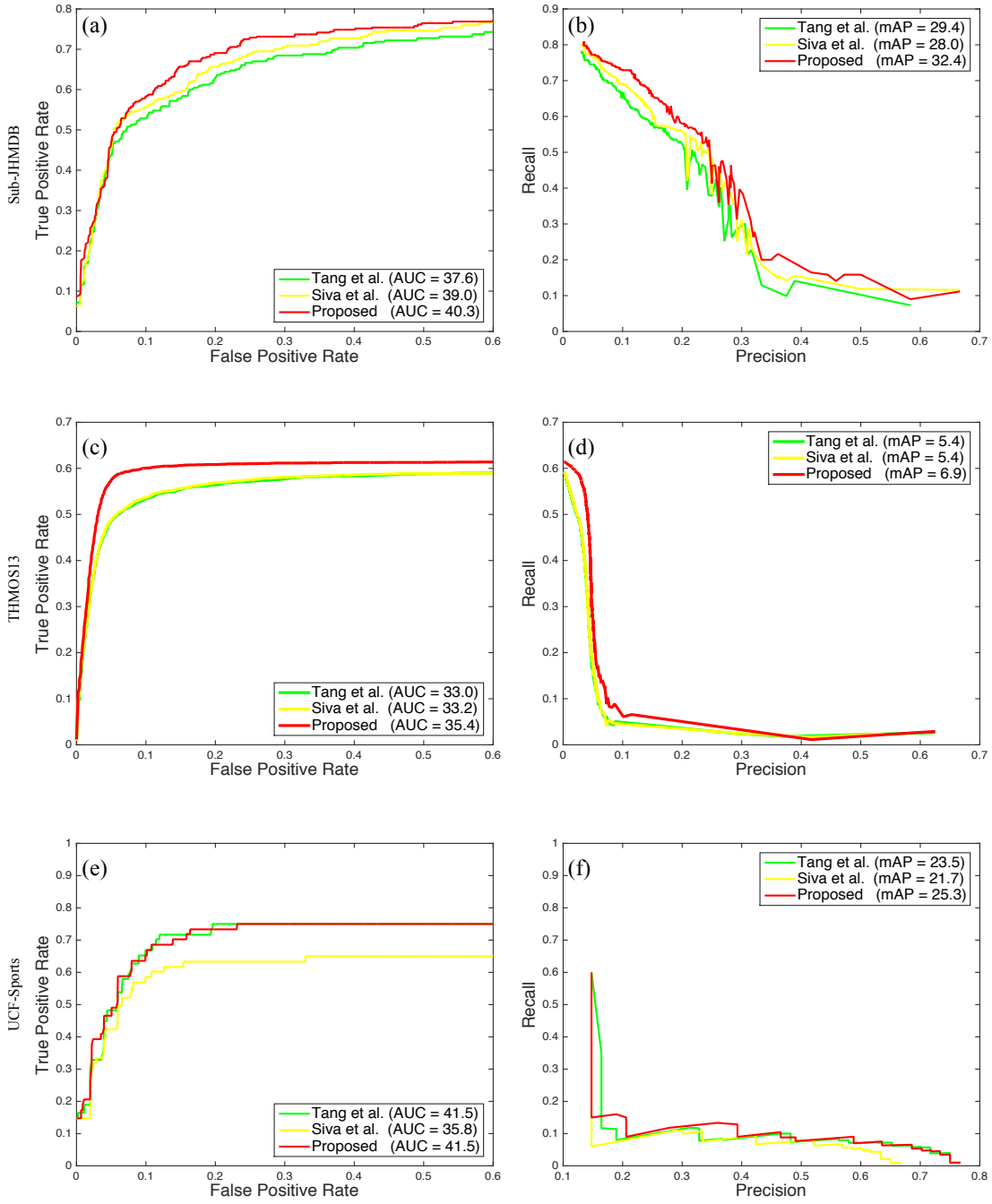

Figure 4.10: Comparsion of the proposed approach, Tang et al. [75] and Siva et al. [63] using ROC, PR , mAP and AUC evaluation metrics. Rows show resuts for sub-JHMDB, THUMOS13 and UCF-Sports respectively. In each row, left figure shows ROC plots and right figure shows PR curves, where AUC and mAP are given in the lagends. The results demonstrate superiority of our approach as compared to baselines.

Table 4.5: Comparison of average precision of each class for proposed and baselines on sub-JHMDB.

Method	**Tang et al.** [75]	**Sive et al.** [63]	**Ours**
Catch	16.3	17.4	**28.0**
Climb stairs	39.3	**39.4**	38.9
Golf	49.0	41.4	**52.3**
Jump	21.0	16.2	**22.6**
Kick ball	24.2	29.0	**31.2**
Pick	19.0	15.3	**23.5**
Pull up	37.5	44.5	**45.3**
Push	23.9	**28.3**	25.9
Run	21.9	22.3	**24.0**
Shoot ball	35.2	33.4	22.5
Swing baseball	17.7	22.44	**24.4**
Walk	30.6	33.2	**49.7**
mAP	28.0	28.6	**32.4**

4.4.5 Analysis and Discussion

Our approach has several components. We quantitatively evaluated each component and show the experimental results for stripped down versions of our method for UCF-Sports in Table 4.6. It can be seen that each component of our method has complementary effects and helps in achieving overall localization accuracy while the fine-grained similarity contributes the most. Fine-grained similarity offers flexible matching among motion patterns between different proposals across multiple videos. It is more robust to abrupt camera motion and actor articulations. Therefore, proposals that cover the same action have high fine-grained similarities and hence get selected through GMCP. We conduct experiments with a different proposal method [27]. Instead of using dense trajectories [81], this method [27] employs hierarchical supervoxel segmentation using motion and color cues. Figure 4.11 shows that our approach has comparable accuracy for both proposal methods on UCF-Sports dataset.

Table 4.6: Component's contribution to overall localization accuracy. First column shows localization obtained using initial action scores only. Second column depicts the same using proposal shape similarity as well. Third and forth column show contribution from global and fine-grained similarity, respectively.

Method	**[I]nitial Score**	**[I]+[S]hape**	**[I]+[S]+[G]lobal**	**[I]+[S]+[G] +Fine-grained**
Diving	64.2	64.2	100	100
Golf Swing	5.50	55.5	66.6	**88.8**
Kicking	20.0	50.0	65.0	65.0
Lifting	0	0	33.0	**83.3**
Riding Horse	33.3	25.0	25.0	**100**
Run	20.0	20.0	40.0	**80.0**
Skateboarding	50.0	41.6	41.6	**91.6**
Swing Bench	10.0	15.0	25.0	**90.0**
Swing SideAngle	38.4	38.4	61.5	**76.9**
Walk	16.6	58.3	25.0	**62.5**
Avg	25.8	36.8	48.3	**83.8**

Ideally, increasing the number of videos should help in getting better annotation, i.e., THUMOS13 which has more than 100 videos per action should have better localization accuracy than UCF-Sports which has on average 15 videos per action. However, the improvement for THUMOS13 is less when compared with UCF-Sports, mainly due to the difficulty level of THUMOS13, as shown in the first row of Table 4.7. The typical behavior of localization accuracy for skateboarding action (THUMOS13) for 100 videos is shown in Table 4.7. The first row in the table shows the mean localization (mean IOU) for the batch of 25 videos. As can be seen, the videos in the 26-50 batch have less localization accuracy as compared to other batches. Employing the proposed method (second row) on the first 1-25 videos boosts their MABO from 24.83 to 29.72. Using proposed method on 1-50 videos (third row) further changes the first batch from 29.72 to 27.82 and second batch from 21.00 to 28.02. A similar pattern can be seen in the third row. It is worth noting that, although localization improves in local batches (1-25, 26-50 and 51-75), the overall localization accuracy drops from 29.72 (25 videos) to 26.26 (75 videos), mainly due to large intra-class variation in THUMOS13 videos.

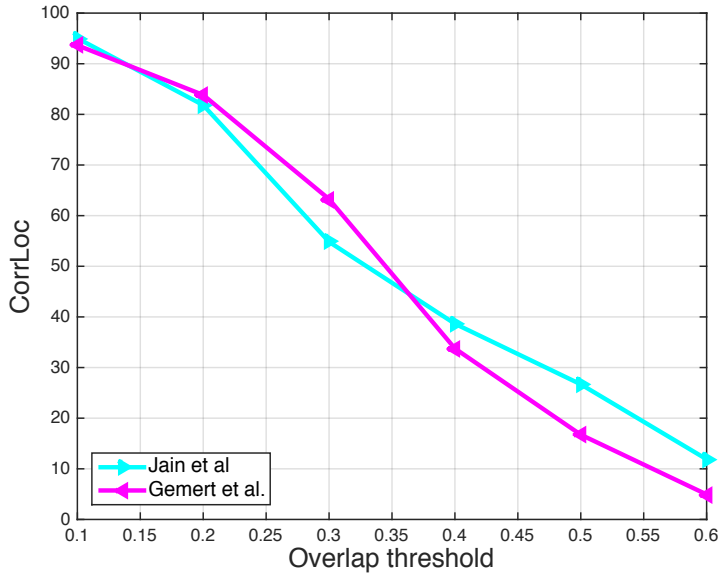

Figure 4.11: Localization accuracy of our approach using different proposal methods.

Table 4.7: Localization accuracy behavior across different batches of videos. The number in brackets shows number of videos used for Localization

Method	**1-25**	**26-50**	**51-75**	**76-100**	**Mean**
Localization	24.83	21.00	25.65	23.75	23.85
Localization(25)	29.72	-	-	-	29.72
Localization(50)	27.82	28.02	-	-	27.92
Localization(75)	27.45	25.43	25.90	-	26.26
Localization(100)	28.29	23.57	26.75	25.91	26.13

Computation Time: We performed experiments on desktop computer Intel Xeon E5620 at 2.4GHz. The following time is for a single instance of MATLAB where the code was executed in a serial manner. The time to compute fine-grained, global, and shape similarities between two proposals respectively are: 3.8×10^{-3} sec , 5.9×10^{-4} sec and 6.1×10^{-5} sec. In order to re-ranked all top proposals GMCP takes 5.83 sec, 5.93 sec and 63.15 min for UCF-Sports, sub-JHMDB and THU-

MOS13 datasets, respectively. For an action containing V=150 videos and P=50 proposal in each video, the time to compute edge weights is (3.8×10^{-3} sec + 5.9×10^{-4} sec + 6.1×10^{-5} sec) $\times$ $V \times (V-1)/2 \times P^2$ = 34.54 hours. Finally, the total time to obtain annotations for UCF-Sports, sub-JHMDB and THUMOS13 is 14, 13 and 691 hours, respectively.

4.5 Summary

We presented a weakly supervised approach to automatically obtain spatio-temporal annotations in a video. In contrast to expensive and time-consuming manual annotations, we obtain these spatio-temporal annotation boxes automatically in a few seconds by matching action proposals across multiple videos using their features and shape similarities. Moreover, we have demonstrated that these annotations can be used to learn robust action classifiers. Note that we intentionally kept our annotation framework generic and independent of action detection approach. Detection methods do not have to use the features and other formulations similar to the ones used in our approach. Any action detection approach can make use of our framework.

Although our method works quite well, it has some limitations to be noted. First, it requires at least a few videos of the same action to find the common pattern across videos and hence obtain action annotations. Second, due to the processing of several videos at the same time, it is not memory efficient. In the next chapter, we discuss our new approach to overcome these limitations.

CHAPTER 5: ACTION LOCALIZATION USING WEB IMAGES

In the previous chapter, we describe a weakly labeled approach which employs multiple videos of the same action to obtain action annotations. However, it is not always possible to have a sufficient number of multiple videos of the same action. Moreover, as discussed earlier, multiple videos do not necessarily increase localization accuracy due to noise in videos. In this chapter, we present a novel approach where we use Web images to obtain action localization in videos.

Recently deep learning has gained lot of interest due to its state-of-the-art accuracy in several computer vision problems involving images and videos. The key factors for deep learning resurgence are availability of millions of image and videos to help training deeper networks which have millions of parameters, and the recent development of parallel processing powers such as GPUs and CPU clusters. One of the limitations of current deep learning systems is their high dependability on thousands of precisely annotated training examples. Although, a few recently proposed methods use weak or unsupervised data; most of the deep learning methods are fully supervised. Annotating thousands of images and videos is costly, requires lots of human hours and contain biases. Therefore, automatic annotation methods are highly required. A good annotation must have two characteristics: it should require least human supervision; and annotation of each video should be independent of annotations of other videos so that we can efficiently do annotations in parallel using GPUs or CPU clusters.

In order to achieve above-mentioned objectives, we develope a new approach to obtain action localization in videos using action images. Human actions are characterized by key poses of a human body. Spatio-temporal locations that contain those key poses and canonical viewpoints of an action can be used to localize an action. Our key idea is to leverage images to find spatio-temporal locations in videos that contain this discriminative information. Figure 5.1 summarizes

our approach.

We first explain image collection and noisy images removal strategy in section 5.1 We then describe action proposal extraction method from images and videos in section 5.2 and 5.3 respectively. In section 5.4, we put forward a new method to construct video proposals using image proposals employing a sparse reconstruction approach with consensus regularization. We describe an extension of our approach to untrimmed videos in section 5.5. Finally, section 5.6 presents experimental results.

5.1 Web Action Image Collection

The first step of our approach is to obtain candidate action proposals in downloaded images. For this purpose, we download images from Internet using the action name such as tennis swing, golf swing etc, as a text query. By searching action names, top retrieved images contain good quality, well-centered action images that capture the key poses and distinct articulation of actions. To avoid ambiguity in a text query, we change some action names slightly, for example using, 'soccer kicking' instead of 'kicking', 'men walking' instead of 'walk'. Although, Google image search quality has been improved significantly over last few years, the retrieved images still contain irrelevant images due to inaccurate query text and polysemy. Figure 5.2 shows some of downloaded images.

We perform the random walk over these images to get rid of image noise. The key benefit of using random walk is that it can discover both small cluster of outliers as well as the images far away from all other images (in feature space) [47]. We define a fully connected graph $\mathbf{Z}(\mathbf{N}, \mathbf{E})$, where $\mathbf{N}$ is the set of all images and $\mathbf{E}$ represents set of edges between them. The weight between any two nodes i and j in the graph is measured by Euclidean distance between their features $\phi(i)$ and

$\phi(j)$, where ϕ represents deep learning features [83], computed over the whole image. Finally, the transition probability between any two nodes i and j is given by

$$p(i,j) = \frac{e^{-\gamma\|\phi(i)-\phi(j)\|_2}}{\sum_{m=1}^{k} e^{-\gamma\|\phi(i)-\phi(m)\|_2}}. \tag{5.1}$$

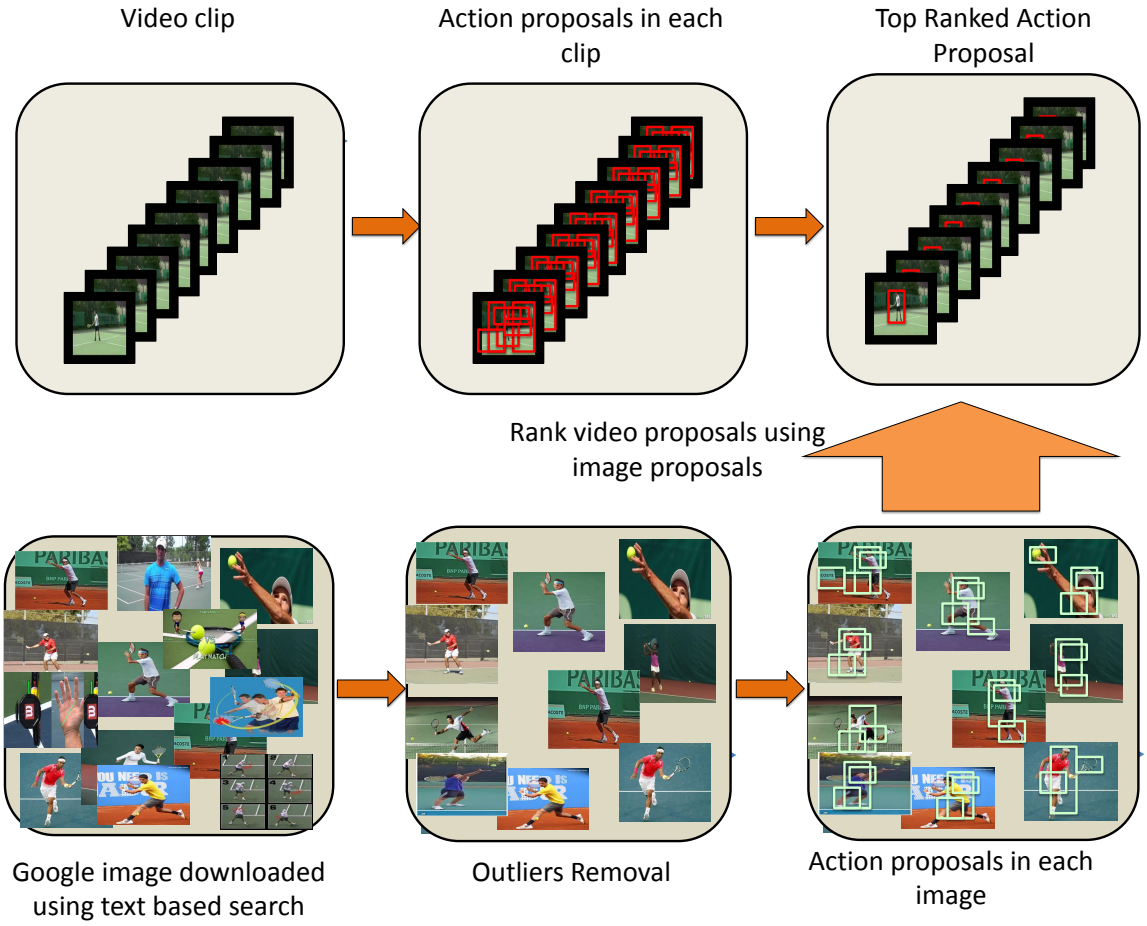

Figure 5.1: This figure illustrates our key idea of action localization in a video using images. We first download images of an action of interest from the Internet. After removing noisy images, we co-localize all the images jointly to obtain action proposals in each of the image. Then, given the candidate action locations in a video, we leverage image proposals to discover the most action representative proposal in a video.

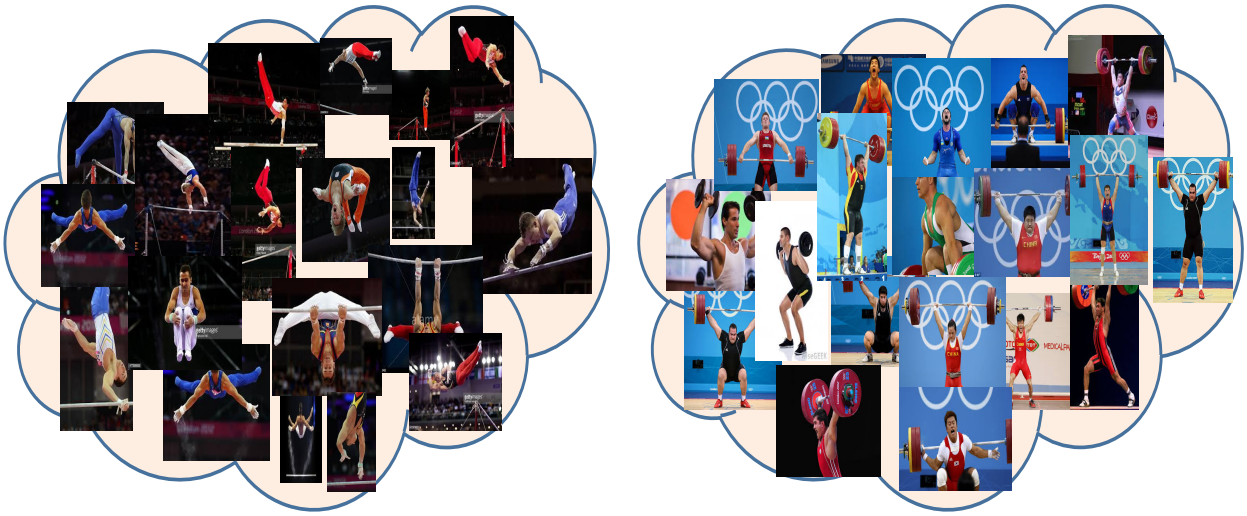

Figure 5.2: This figure shows some of downloaded images for swing side-angle (left) and weight lifting (right).

The random walk over the graph is then formulated as:

$$r_k(j) = \beta \sum_i r_{k-1}(i) p_{ij} + (1 - \beta) v_j, \tag{5.2}$$

where $r_k(j)$ represents relevance score of the image j at k^{th} iteration, v_j is its initial probabilistic score for being non-noisy and β controls the contribution of both terms to the final score. Due to the absence of any prior knowledge about images, we assign the same initial probabilistic score to all the images. The relevance score $r_k(j)$ is iteratively updated for all nodes until fixed number of iterations are achieved. The images with low relevance score can be considered as outliers and subsequently removed. In our experiments, we removed 30% of the originally downloaded images. When removing images more than 30%, we start losing good quality images. In experiments, we use β=0.99. Some of the typical images removed by random walk are shown in Figure. 5.3.

Figure 5.3: Noisy golf swing images removed by the random walk. These images include cartoons, people in unusual backgrounds and clipart. Rightmost image in the second row represents the failure case, which random walk is unable to remove (perhaps due to its similarity to golf swing in the feature space).

5.2 Action Proposals in Images

Although images downloaded using the text query belong to the same overall concept; they are mostly captured in different scenes and contain distracting backgrounds. Using these images naively is detrimental to video proposals ranking (see Table 5.1). To get rid of unnecessary backgrounds, we propose to localize the action in images. To localize the action in the downloaded images, we apply recently proposed state-of-art unsupervised localization method [8]. We use this method because of its excellent performance on many complex datasets [14].

Following [8], we extract hundreds of candidate action proposals [46] from each image. The objective is to obtain the proposals which represent the most common concept (the action in our case) across all the images. To achieve this, we efficiently match action proposals across all the images using Probabilistic Hough Matching (PHM) [8]. PHM incorporates appearance and geometric consistency between patches. Specifically, inspired by hough transform, in PHM, matching between multi- scale proposals casting vote for each other and ultimately obtain high confidences.

The PHM is performed on local regions within proposals by carefully considering their scale and localization variations. The score of a local region in proposal p_m with respect to p'_m is given as:

$$\psi(p) = \max_{r'} c((r, r')|(p_m, p'_m)), \tag{5.3}$$

where c represents Hough matching confidence of local region r in p_m with respect to p'_m.

The high region score represents the highest matched proposal across images. However, it does not provide the explicit action localization as the background regions can also have good matches. Therefore, we use both the standout score [8] of each proposal and the PHM based region matching to obtain the final action localization in images. In our experiments, we use only top two action proposals from each image. Selecting more than two proposals increases the computational time of next steps, while not always helping the performance. Figure 5.4 shows automatically generated images' proposals for actions of THUMOS14. The leftmost images in the second row show the failure cases, where we are unable to localize the action of interest.

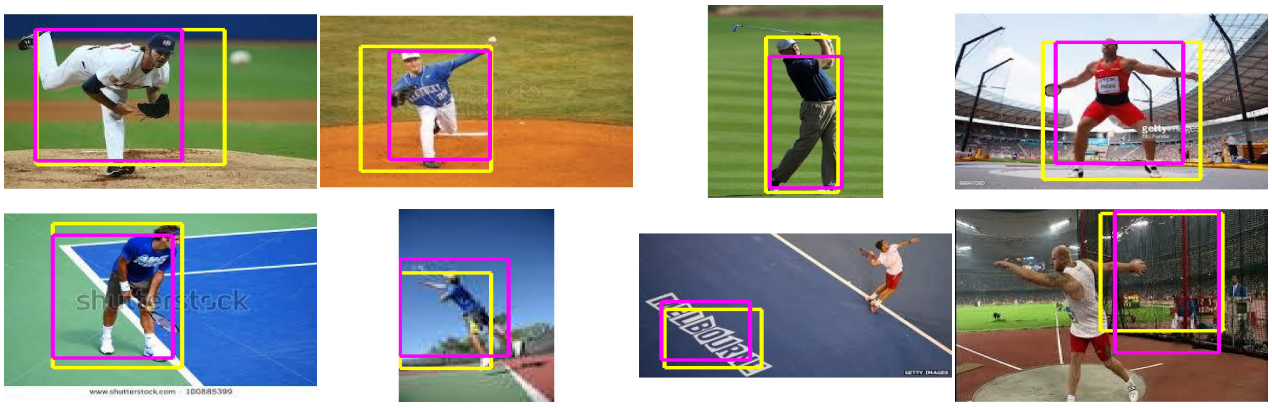

Figure 5.4: Automatically generated action proposals in images. In the bottom row, last two images (from right) show the failure cases due to very small size of actor and cluttered background.

5.3 Action Proposals in videos

Our end goal is to obtain spatio-temporal action localization in video using image action proposals generated in the previous section. Therefore, we first estimate candidate action locations in the video and try to remove the majority of camera and background generated proposals. In this work, we employ supervoxel segmentation based approach to generate action proposals [49]. However, our method does not depend on specific action proposal methods and any action proposals method can be used [27, 81]. Following [49], we compute fixed number of superpixels from each video frame and estimate mean color, color histogram and optical flow histogram within each superpixel. Given n number of superpixels, we build a graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is a set of superpixels and $\mathbf{E}$ represents a set of edges between them. We use discontinuity preserving first and second order spatial edge weights between superpixels n and m, where the first order edge weight is given by:

$$\begin{aligned} e^{nm,s} = \alpha_1 d_1(n,m) + \alpha_2 d_2(n,m) + \alpha_3 d_3(n,m) \\ + \alpha_4 d_4(n,m) + \alpha_5 d_5(n,m), \end{aligned} \tag{5.4}$$

where d_1 corresponds to distance between color means, d_2 and d_3 represent distance between color and flow histograms and d_4 and d_5 represent geodesic distance between superpixel centroids computed through motion and color boundaries.

In addition to spatial edges, we also build temporal edges [49] given as

$$e^{nm,t} = \alpha_7 d_1(n,m) + \alpha_8 d_2(n,m) + \alpha_9 d_3(n,m), \tag{5.5}$$

where d_1, d_2 and d_3 are the same as described before and m and n represents temporal neighbors. Hierarchical clustering on this graph results into supervoxels segmentations. Finally, action proposals are built by merging supervoxels using randomized Prim's maximum span tree algorithm

[46], extended to videos. During proposals generation, appearance, motion and size similarities of superpixels are taken into account. Typical examples of few action proposals for UCF Sports videos are shown in Figure 5.5. Although the above method generates significantly less number of action proposals (approx. 2000 in each video), their number is still huge for our application, since we want to obtain only *the most action representative proposal* in each video clip.

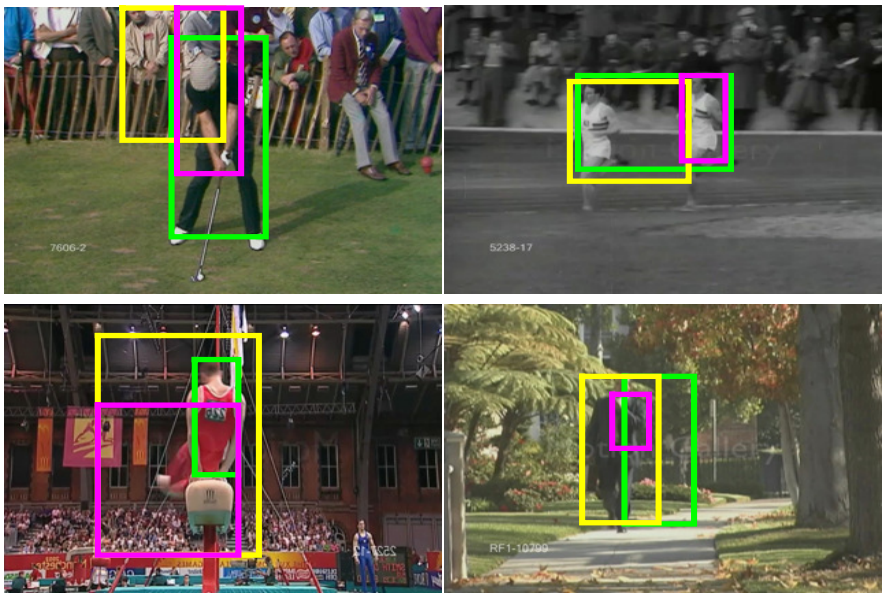

Figure 5.5: Video action proposals. Colors in the figures are randomly assigned.

Human actions are mainly characterize by motion. We use this important cue for two purposes. First, we use it to discard camera and background generated proposal (as they would have small optical flow gradients). Secondly, we use it to facilitate action proposal ranking. To this end, we employ optical flow gradients within each video proposals. We first compute Frobenius norm of optical flow within each proposal, as defined is Eq. 4.1. The motion score, η_p, of each video proposal, p_v, is then defined as weighted summation of frobenius norm, namely,

$$\eta_p = G_l(x_c, y_c) \times G_s(h, w) \times \sum \|U_X\|_F \,, \tag{5.6}$$

where x_c, y_c, h, w represent center coordinates, height and width of the proposal respectively and $\|U_X\|_F$ represents frobenius norm of optical flow. Gaussians G_l and G_s encourage proposals that are in the center of the video and are in vertical shapes since humans in these action videos are mostly in the center and are in upright position.

Assuming η_p, as a detection score, we perform Non-Maximal Suppression (NMS) to obtain a few proposals, which have high optical flow gradients and have a small overlap with each other. In our experiments, we keep at most fifty proposals from each video. This results in a huge decrease in the computation for further steps. Finally, we normalize motion score η_p of all proposal within a video between zero and one. We use this normalize score to represent motion saliency of each proposal.

5.4 Ranking Video Action Proposals using Image Action Proposals

In this section, we present our key idea of ranking video action proposals, $\mathbf{P_v}$, using image action proposals, $\mathbf{P_m}$. We achieve this by reconstructing video action proposals as a linear combination of image action proposals. The main idea is that video action proposals which can easily be reconstructed using image action proposals (i.e., have low reconstruction error) can be considered to be capturing the key poses and viewpoints of the specific action and therefore represents the action of interest.

Suppose a video contains k number of video action proposals, $\mathbf{P}_v = [p_v^1, p_v^2, \ldots, p_v^k]$. Within each proposal, we extract VGG-16 visual features [83] from each of the key frame (bounding box). Let $\mathbf{\Pi^f} \in \mathbb{R}^{d \times n}$ represents the matrix obtained by vertical concatenation of all key frames features within a proposal, where d is the dimension of visual feature, and n is the number key-frames within proposal. Similarly, $\mathbf{\Upsilon^f} \in \mathbb{R}^{d \times m}$ represents a vertical concatenation of visual features from

all image proposals, where m represents the total number of image proposals.

The straightforward approach would be to reconstruct each of the video proposal bounding box independently using image proposals and aggregate the reconstruction error for all the bounding boxes to obtain overall proposal action score. Although appealing, it ignores the underlying temporal structure of the video. Videos are not just the collection of frames but the sequence of frames and hence contain temporal information. Therefore, we propose to reconstruct all proposal bounding boxes jointly using the constraints that push the coefficients for each bounding box towards a common consensus, thus enforcing the coefficient similarity across multiple frames. Moreover, we introduce sparsity constraint to take care of noise in image data. Consensus regularization has been introduced recently for different applications [9, 93].

To achieve above goal, we minimize following the objective function:

$$\mathbf{Z} = \min_{\mathbf{C}} \left\| \mathbf{\Pi^f} - \mathbf{\Upsilon^f C} \right\|_{\mathbf{F}}^{\mathbf{2}} + \lambda_{\mathbf{1}} \left\| \mathbf{C} - \bar{\mathbf{C}} \right\|_{\mathbf{F}}^{\mathbf{2}} + \lambda_{\mathbf{2}} \left\| \mathbf{C} \right\|_{\mathbf{1}}, \tag{5.7}$$

where the first term minimizes reconstruction error and second and third term enforce consistency (across columns) and sparsity in coefficient matrix $\mathbf{C}$, respectively. The consensus matrix $\bar{\mathbf{C}}$ is obtained by columns-wise concatenation of mean of the coefficient matrix $\mathbf{C}$.

We solve the optimization mentioned in Eq. 8 using a variant of two-metric projection algorithm [19, 57]. We divide the optimization variables c_i into two sets: active set and working set. Active set, $\mathcal{A}$, contains the variables that have positive partial derivative and are close to zero.

$$\mathcal{A} = \{i | c_i < \epsilon, \nabla_i \mathbf{Z}(\mathbf{C}) > \mathbf{0}\} \tag{5.8}$$

Similarly, variables that have negative partial derivative or that are sufficiently non-zero belong

to working set, $\mathcal{W}$. We compute a diagonally-scaled projected pseudo-gradient step for active set variables and a projection of Newton step along working set, namely,

$$\begin{aligned} \mathbf{C}_{\mathcal{W}} &\leftarrow \mathcal{P}[\mathbf{C}_{\mathcal{W}} - \beta \mathbf{H}_{\mathcal{W}}^{-\mathbf{1}} \nabla_{\mathcal{W}} \mathbf{Z}(\mathbf{C})], \\ \mathbf{C}_{\mathcal{A}} &\leftarrow \mathcal{P}[\mathbf{C}_{\mathcal{A}} - \beta \mathbf{D}_{\mathcal{A}} \nabla_{\mathcal{A}} \mathbf{Z}(\mathbf{C})], \end{aligned} \tag{5.9}$$

where $\mathcal{P}$ is orthant projection and $\mathbf{H}$ is Hessian matrix. Note that, given positive diagonal scaling matrix $D_{\mathcal{A}}$, combined gradient direction is descent, unless $\mathbf{C}$ is optimal. We iteratively solve the above equations until we obtain the optimal solution or the maximum number of iterations are met.

We optimize Eq. 8 for each proposal in the video clip and estimate the reconstruction error. We normalize reconstruction errors of all proposals within a video between 0 and 1. The final action score Λ_p of each proposal, p_v, is simply given as:

$$\Lambda_p = (1 - \mathcal{R}_p) + \eta_p, \tag{5.10}$$

where $\mathcal{R}_p$ and η_p represent reconstruction error and motion saliency (calculated in Section 4) of proposal, p_v.

Note that we have experimented with several state-of-art domain adaptation methods such as [22, 26], however, either they do not help at all or have a diminishing effect on the performance.

5.5 Action localization in Untrimmed videos

The proposed approach is generic in nature and can be applied to any action dataset including recently introduced extremely challenging untrimmed action datasets such as THUMOS14 [31]. This dataset contains long YouTube sports videos, mostly gathered from news and documentaries.

The general trend in these videos is that they contain: newscaster or reporter, clips showing the crowd and stadium, people talking about the specific sport and finally the actual action clips somewhere in between these irrelevant clips.

To use our approach on untrimmed videos, we first divide long videos into shots [2]. We start with the assumption that each shot contains an action. By considering each shot as a $trimmed$ video, we compute top ranked action proposal from each video using exactly the same procedure as described in Section 3, 4 and 5. After computing the most representative action proposal in each shot (Section 5.3), we compare the action score (Eq. 5.11) of these top ranked proposals across the shots. Intuitively, the shots that contain an action would have top ranked proposals with high action score as compared to the shots that do not contain action. We max-normalize the reconstruction error of shots across the video. By sweeping the threshold of reconstruction error, we generate ROC curve as shown in Figure 5.6.1.

5.6 Experimental Results

The objective of our experiments is to quantitatively evaluate the performance of proposed approach, verify that each component contributes to its final accuracy and demonstrate the generality. To this end, we performed extensive experiments on trimmed as well untrimmed action datasets. For shot detection, we compute RGB histogram of frames as a feature representation.
For all other experiments, we use CNN features [83], computed within image/video proposals bounding boxes.

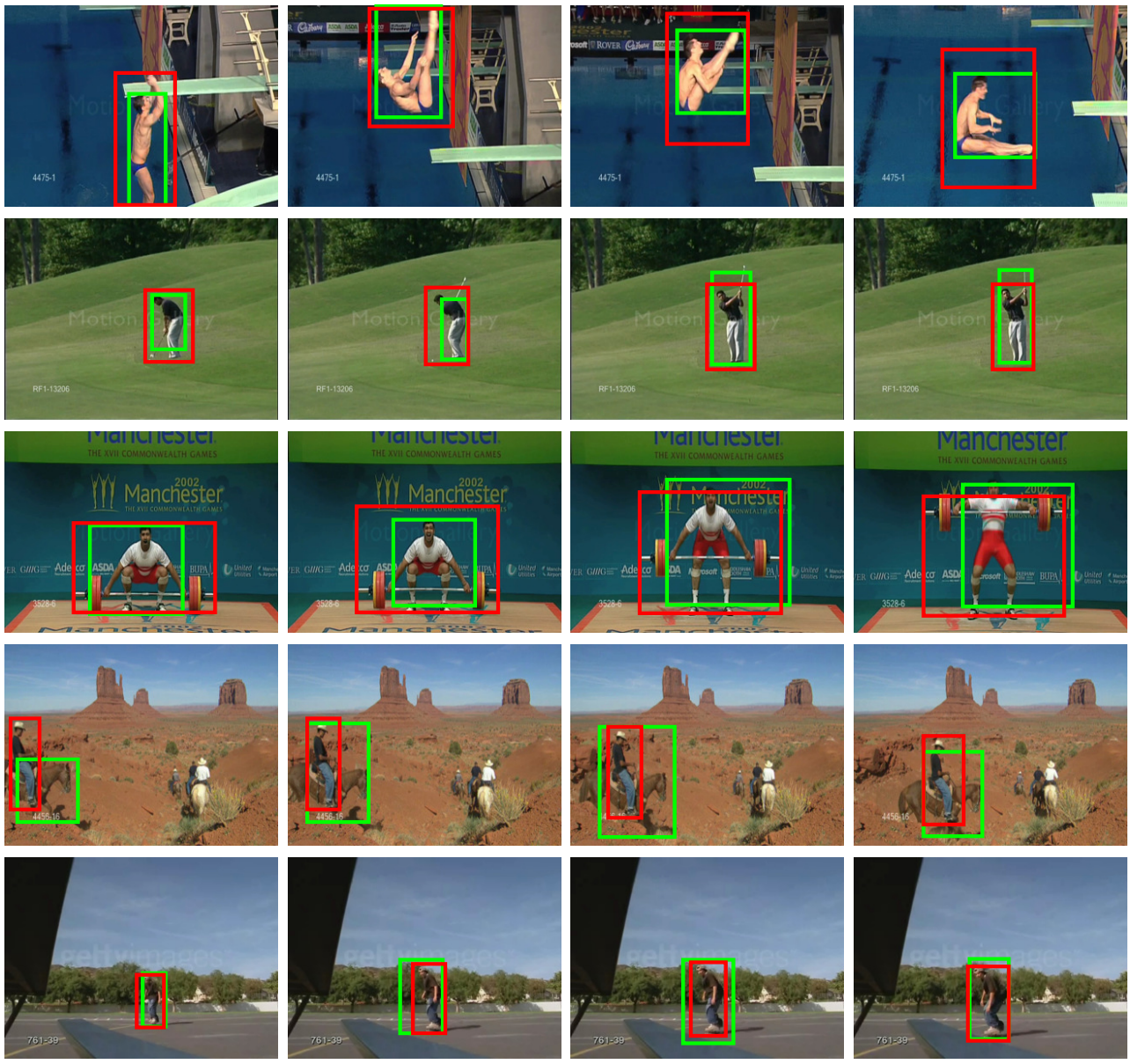

Figure 5.6: Localization results (Top-ranked proposal) from UCF-Sports for five actions. We show four frames of each action video. Red box indicates ground truth and green box shows localization results.

Figure 5.7: Localization results (Top-ranked proposal) from UCF-Sports. for five actions.We show four frames of each action video. Red box indicates ground truth and green box shows localization results.

5.6.1 Experiments on Trimmed Action Dataset

For experiment on trimmed dataset, we have chosen UCF-Sports [54] and THUMOS13 [67] because of there complexity and that several recent works have used in their experiments [81, 27, 76].

Table 5.1: Quantitative results for UCF-Sports. First column shows localization accuracy of reconstructing video proposals using all images (including noisy ones). The second column shows the same after noise removal using random walk. The third column shows localization accuracy of reconstructing video proposals from image proposals without enforcing sparsity and consensus constraints. Localization accuracy of complete reconstruction model (Eq.8) is shown in fourth row. Finally, fifth column shows accuracy of complete method. The results indicate that noise removal, image proposals, regularization and motion saliency; all contribute to overall localization accuracy.

Method	**Images w/o noise removal**	**Images w/ noise removal**	**Image Prop w/o constraints**	**Reconstruction error**	**Complete Method**
Diving	78.57	78.75	85.71	92.86	100.00
Golf Swing	77.78	83.33	83.33	100	94.44
Kicking	60.00	65.00	80.00	85.00	85.00
Lifting	100	100	100	100	100
Riding Horse	75.00	75.00	100	91.67	100
Run	61.54	61.54	69.23	92.31	84.62
Skateboarding	50.00	50.00	66.67	88.33	83.33
Swing Bench	80.00	85.00	100	95.00	100
Swing Sideangle	7.69	30.77	61.57	69.23	84.62
Walk	61.90	52.38	90.48	76.19	95.24
Average	65.25	68.16	83.70	88.56	92.72

In these datasets, the action spans the complete video clip. These broadcast videos contain large camera motion, cluttered background, variable viewpoints, and occlusion. UCF-Sports dataset contains 150 videos and include 10 actions including diving, golf swing, kicking, lifting, horse riding, running, etc. THUMOS13 localization dataset contains 24 human actions that have spatio-temporal annotations. These actions include cricket bowling, biking, salsa spin, etc. This dataset contains 3207 videos. We use all videos of both datasets for evaluation (Except Walk-Front-005 in UCF-Sports since it is actually a running action).

To evaluate localization accuracy, we use the standard intersection over union metric at 20% threshold [27, 76]. The localization accuracy of our complete method for UCF-Sports is given in Table 5.2. We compare our method with two strong baselines: CRANE [75] and Negative Mining [63].

Table 5.2: A comparison of our approach with related weakly supervised annotation methods on UCF-Sports

Method	**CRANE** [75]	**NML** [63]	**Ours**
Localization	65.41	63.01	**92.72**

Similar to the proposed approach, both of these techniques are weakly supervised annotation methods, i.e., they only assume video level labels. The comparison shown in Table 5.2 indicates the significantly improved localization accuracy of our method. Note that we use the same features [83] for all three methods.

Table 5.3: Localization accuracy of UCF-Sports and THUMOS13 (24 classes) at various thresholds.

Threshold	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**
UCF-Sports	93.9	92.7	82.1	61.0	40.7	18.5
THUMOS13	78.0	62.7	47.8	28.8	13.8	4.6

In Figures 5.6 and 5.7, we show qualitative examples of localization on UCF-Sports dataset. We show four frames for a video from each action. It can be seen that our method performs quite well despite large camera motion (diving, kicking), scale changes (walking), cluttered background (horse riding, skateboarding), small actor size (running, golf swing) and abrupt motion (swinging).

Our method contains several components. We evaluate the contribution of each component towards final localization accuracy in Table 5.1. First column indicates localization accuracy, where we use all of the downloaded images (without removing noisy ones) in our reconstruction framework. Removing noisy images gives 3% improvement in localization accuracy (second column). Reducing the effect of images background noise through proposal, we achieve further 15% improvement. By enforcing consistency and sparsity in coefficient vectors among multiple frames of the proposal,

we obtain 5% improvement. Finally, by adding motion score, we achieve further 5% improvement. Our results demonstrate that each component of our approach is necessary and contributes towards final localization accuracy. Moreover, our results reinforce that the web images do have the ability to make a significant impact on action localization in videos.

Table 5.3 shows localization accuracy of top ranked proposals in UCF-Sports and THUMOS13 across different overlap thresholds.

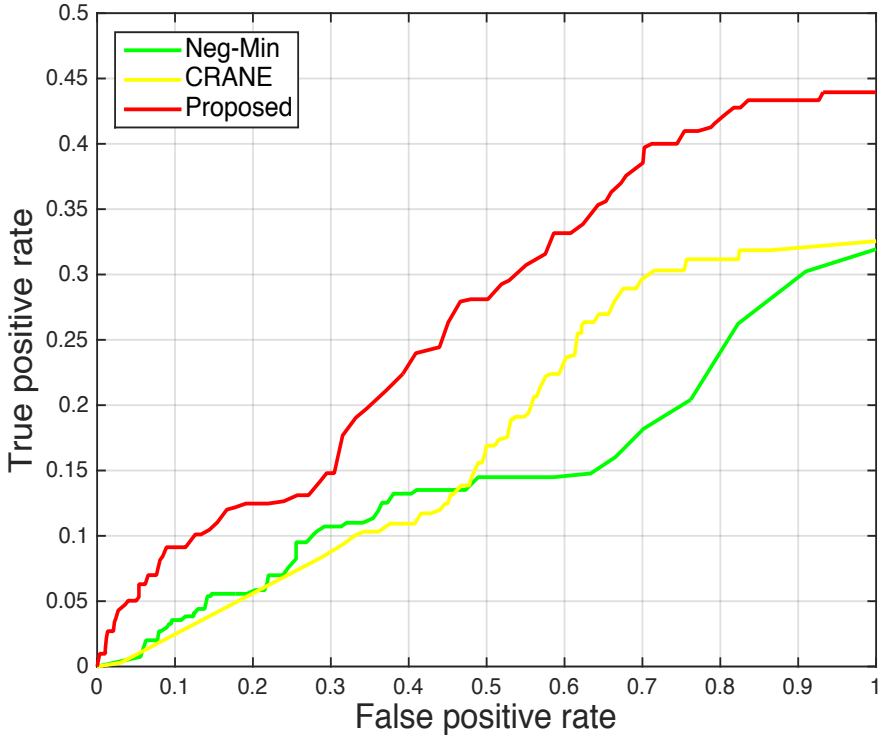

Figure 5.8: Mean ROC curves for four actions of THUMOS14: Tennis swing, Golf swing, Throw Discus, and Baseball pitch. The results are shown for Negative Mining approach [63] (green), CRANE [75] (yellow) and Proposed method (red).

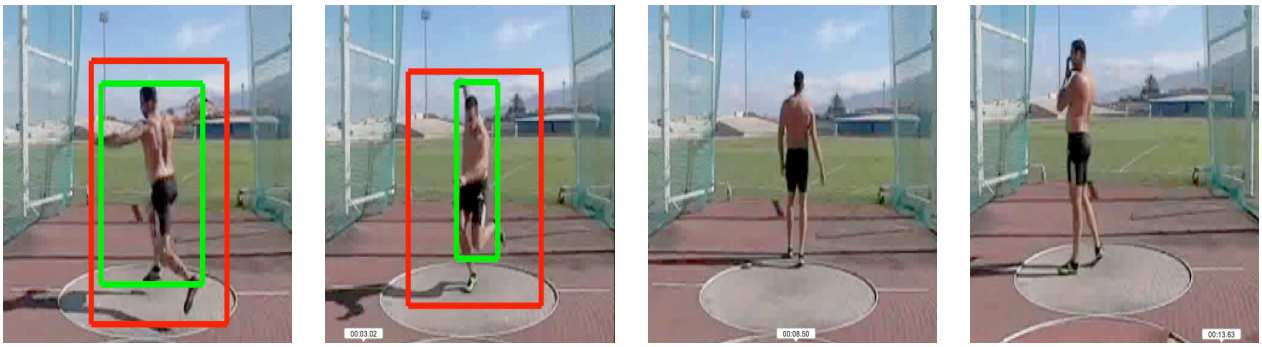

Figure 5.9: Throw discus localization results at different instances of time. Note that for the last two frames, actor is not performing throw discuss.

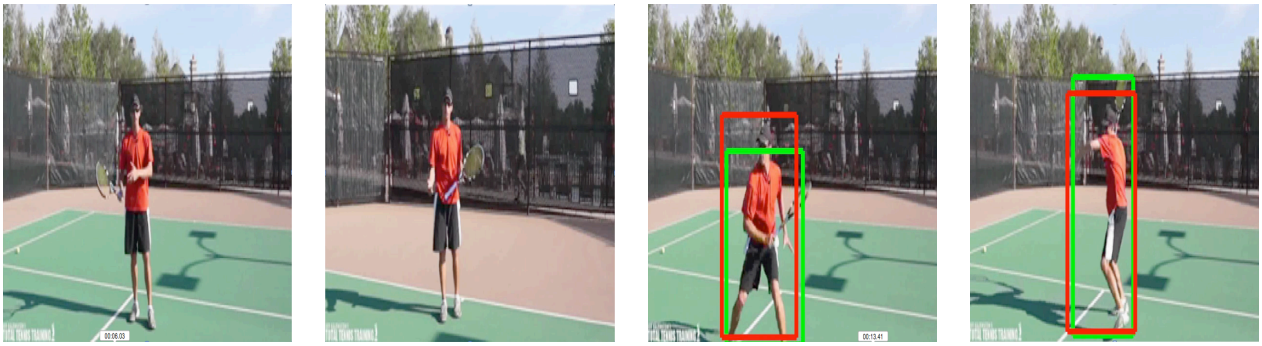

Figure 5.10: Tennis swing results at different instances of time. Note that for the first two frames, actor is not performing tennis swing.

5.6.2 Experiments on Un-Trimmed Action Dataset

To demonstrate the effectiveness of the proposed approach, we have evaluated it on a part of recently released un-trimmed action dataset [31]. This dataset was released in 2014 in THUMOS challenge workshop. In addition to having the cluttered background, severe occlusion, and huge camera motion, these extremely challenging real-world videos contain several irrelevant frames

such as non-action frames and multiple instances of the same action.

THUMOS14 test-set contains 20 actions, where only temporal annotations are provided *without any spatial annotations*. To evaluate spatio-temporal localization accuracy of our method on this dataset, we manually annotated four actions: baseball pitch, golf swing, tennis swing and throw discus. Specifically, we annotated around 35, 000 video frames (these annotations are publicly available on project web page). Baseball pitch, golf swing, tennis swing and throw discus contain 40, 141, 80, and 28 number of action instances, respectively.

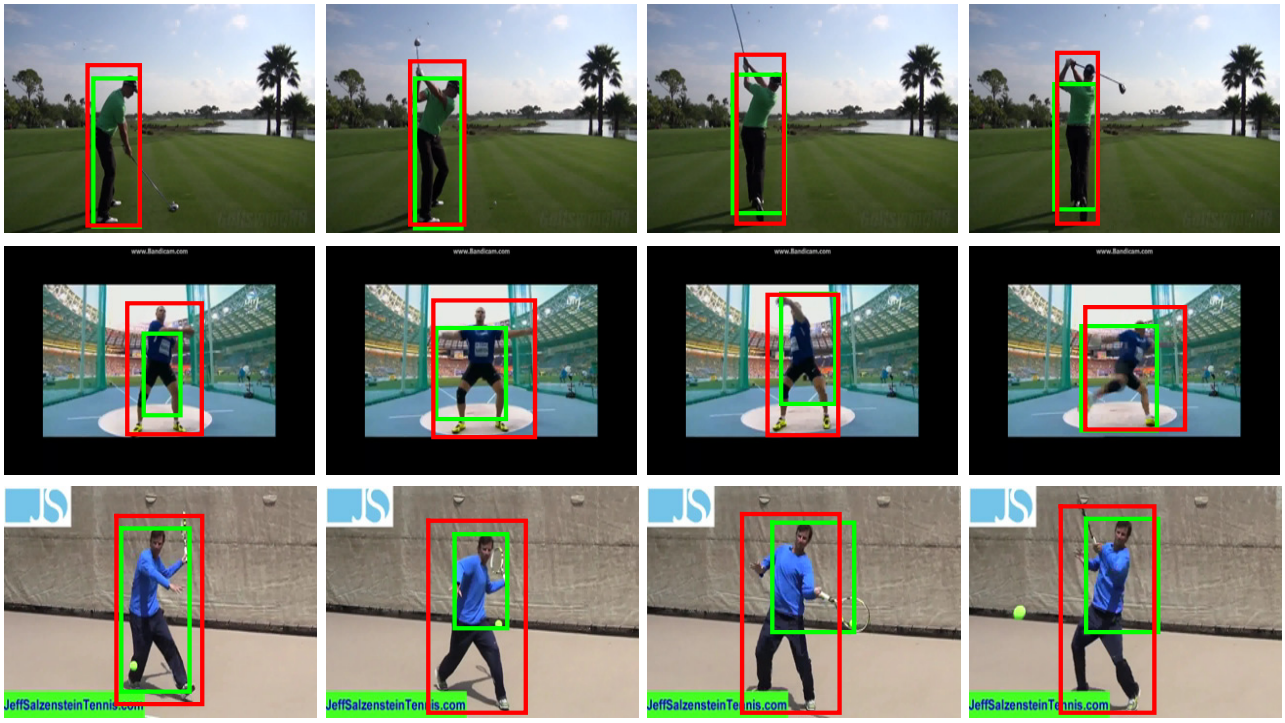

Figure 5.11: Localization results for THUMOS14. We show four frames of each action video. Red box indicates ground truth and green box shows localization results.

Given a video, we first divide it into shots or clips. We, then, compute video action proposals within each clip by assuming that each clip contains the action. We compute the action scores of all proposals within the clip and obtain the most action representative proposal in every shot.

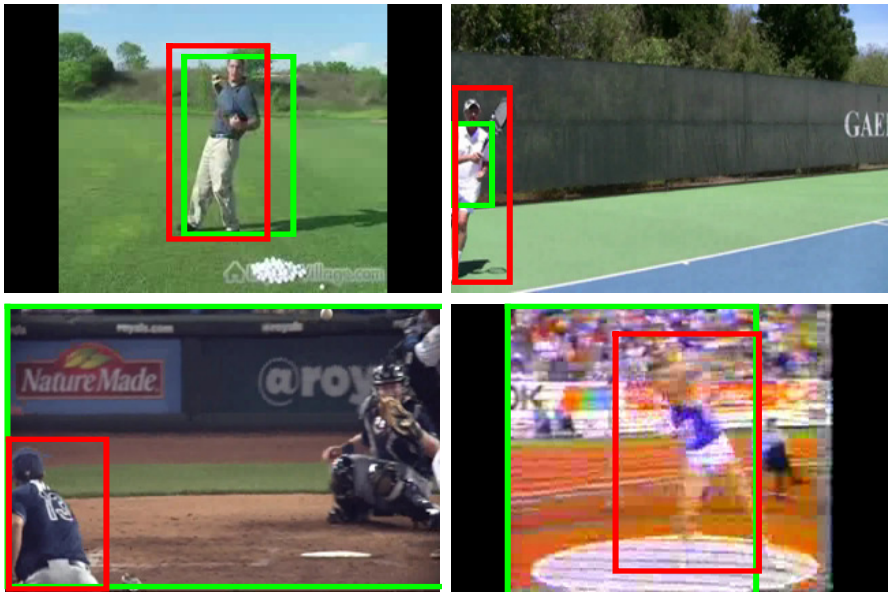

Figure 5.12: Failure cases on THUMOS14 dataset

We consider the action score as an action detection score and evaluate localization accuracy using intersection over union metric at 10% threshold. The ROC curve for all four actions is shown in Figure 5.6.1. Again, we compare our results with weakly labeled localization methods [75, 63] and obtain better results for all four actions. Improved results as compared to strong baseline methods signify the effectiveness of the proposed approach. We use lower threshold criterion due to the extreme difficulty of the dataset. Even though the results of all three methods are lower as compared to state-of-art results on similar actions in trimmed datasets, we consider these results encouraging, due to the complexity of dataset. The qualitative results for all four actions are shown in Figures 5.9, 5.11 and 5.10.

Figure. 5.12 shows some of the typical failure cases on THUMOS14 dataset. The figure on the top-left shows a frame from golf swing video. In this video of more than 5000 frames, the complete golf swing action happens only for 500 frames. In the rest of the video, the person is teaching golf swing techniques and performing in-complete golf swing action several times. Although we

achieve good localization over the actor, our method has the problem in distinguishing complete action from the incomplete ones. Other failures occurred due to actor's occlusion and blurred video.

5.7 Summary

In this chapter, we presented a new approach to spatio-temporally localize an action in a single video. As compared to previous similar works, we do not assume the availability of multiple videos, prior annotations or clean images. Our experimental results show that impressive action localization can be achieved by reconstructing candidate action locations by leveraging freely available Internet images. Our framework tackles noisy images through the random walk and sparse representation, removes background and camera generated video proposals through optical flow gradients and preserves the temporal smoothness of video by enforcing consistency of coefficient vectors across multiple frames. Our extensive experiments on trimmed as well as un-trimmed action datasets validate the effectiveness of proposed ideas and the framework.

Note that the method discussed in this chapter rely on action proposals methods. However, what if we do not have good quality action proposals?. In the next chapter, we present a novel approach to obtain a few good quality action proposals.

CHAPTER 6: ACTION PROPOSAL RANKING THROUGH PROPOSAL RECOMBINATION

In the previous chapter, we describe a weakly labeled approach for action localization using web images. One of the underlying assumptions in that approach is the assumption of the availability of video level annotations for searching corrosponding action images. However, with the recent explosive growth of action datasets [31], it is not easy to obtain video level annotations. Furthermore, the methods proposed in previous chapters use action proposals. Since these methods use hierarchical segmentation or clustering, they tend to generate thousands of action proposals in each video, however, only a few of them contain action. Moreover, while most object proposal methods [79, 39, 101, 1] produce objectness scores (or ranking) for the proposals; however, most action proposal methods [27, 81, 49] do not produce actionness scores (or ranking). For efficient action detection system, we need action proposal method which can generate a few properly ranked good quality action proposals so that action classifier can be tested on only a few locations. This will not only increase the efficiency of action detection system but also improve accuracy by reducing false positive rate.

In order to achieve above-mentioned objectives, we propose a new approach to obtain action proposals through sub-proposals recombination. Our approach needs neither bounding box annotations nor video level labels. Moreover, due to the recombination of sub-proposals across different proposals, it has the ability to discover new proposals that can be better than all of the initially generated proposals. Our method is summarized in Figure 6.1.

6.1 Action Proposal Recombination and Ranking

The proposed approach for action proposal ranking starts by obtaining candidate action proposals and then recombines them in order to get fewer but better-ranked proposals. Since the number of candidate action proposals is huge in number and many proposals are noisy, in order to obtain robust ranking, we divide each proposal into sub-proposals and build a graph across the proposals with the sub-proposals as the nodes. The node score of the graph is a combination of image-based actionness score and motion score. Edges between nodes impose the frame consistencies, and their scores are a combination of intersection-over-union and appearance similarity between sub-proposals. The action proposals are generated and ranked in an iterative manner: to maximize the node+edge scores.The combined node+edge score is assigned to each proposal as its score. After selection of this proposal, all related nodes are removed from the graph and the next proposal is selected in the same way. This is an iterative process and it can produce and rank an arbitrary number of action proposals as needed. Note that our method is not limited to any specific methods for generating candidate action proposals; any recent action proposal method [96, 49, 27, 81] can be employed within the framework. In experiments, we demonstrate the superior performance of our approach using initial proposals obtained from three different proposal methods.

In what follows, we introduce the graph formulation in Section 6.2, and explain the node and edge scores, respectively, in Sections 6.2.1 and 6.2.2.

6.2 Graph Formulation

We formulate the problem of action proposal ranking as a graph optimization problem (Fig. 6.1). The graph $G = \{V, E\}$ consists of a set of nodes $V = \{v_i^f | i = 1..N, f = 1...F\}$ and edges $E = \{e_{i,j}^f | i, j = 1...N, f = 1...F\}$, where i, j are the proposal indices, f is the sub-proposal

index, N is the number of action proposals and F is the number of sub-proposals in the video. In order to obtain sub-proposals, we divide video into five equal temporal segments. Note that the accuracy of our approach is stable across several different number of sub-proposals.

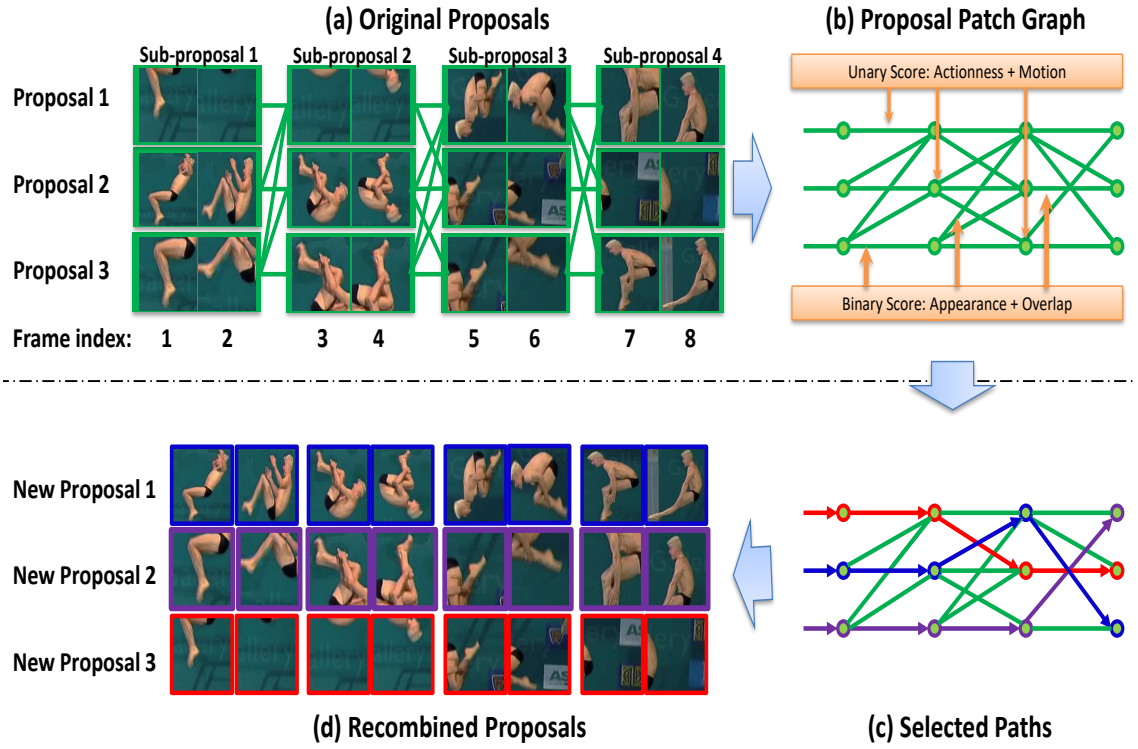

Figure 6.1: An illustration of the proposed method. In this illustration, there are 3 action proposals with 8 frames each. (a) shows the sub-proposal patches of the original action proposals. (b) shows the corresponding graph. Each node in (b) represents one sub-proposal, and edges represent the consistencies between the nodes. (c) shows the 3 top selected paths in the graph (in the order of blue, red, and purple). (d) shows the ranked new action proposals corresponding to the graph in (c), and it is easy to see that these are much better than the original proposals in (a).

Node scores are defined as

$$\Phi = \lambda^i \cdot \Phi^i + \lambda^m \cdot \Phi^m, \tag{6.1}$$

where Φ^a and Φ^m are the image-based actionness score and motion score respectively. λ^a and λ^m are the corresponding weights for each term. The edge scores are defined as

$$\Psi = \lambda^o \cdot \Psi^o + \lambda^a \cdot \Psi^a, \tag{6.2}$$

where Ψ^a, Ψ^o are the appearance similarity and shape consistency scores, and λ^a, λ^o are the weight adjustments accordingly. Combining the node and edge scores we get as follows:

$$E(P) = \sum_{f=1}^{F} (\Phi^f_{p_f} + \lambda \cdot \Psi^f_{(p_f, p_{f+1})}), \tag{6.3}$$

where $P = \{p_1, ..., p_F\}$ are the sub-proposal indices selected to make one proposals, Φ^f_i is the node score of ith proposal in f sub-proposal, $\Psi^f_{(i,j)}$ is the edge score for the edge that connects two ith and jth proposal in temporally adjacent sub-proposals, and $\Psi^F_{(.,.)} = 0$ (i.e. just to ensure the notation to be consistent for the last frame). The goal is:

$$P^* = \arg\max_P E(P). \tag{6.4}$$

Our aim now is to select optimal paths through this graph. The graph G in Fig. 6.1 is a directed acyclic graph and the solution for Eqn. 6.4 can be obtained in polynomial time using dynamic programming [94], we use the value of this function to rank different proposals. After each iteration, the selected nodes are removed, and the next subset of nodes is selected using the same process until a specific number of action proposals are obtained.

Each term of the node score Φ in Eqn. 6.1 and edge score Ψ in Eqn. 6.2 is introduced in next two sections.

6.2.1 Node score

Below, we describe each component of our node score.

6.2.1.1 Image-based actionness score

The objective of the image-based actionness score is to provide a probability score for each proposal patch to measure whether or not some action is being performed there. In contrast to generic object-ness, little work has been done for the generic image-based actionness score (also sometimes called actionness). It is well known that different actions share much high-level articulation and semantic information. Many actions, for instance, walking, jogging, running, kicking, diving, swinging, share similar patterns of motion in different images of the videos. Therefore, we believe that learning a generic image-based action detector for all actions can robustly provide us a probability of the presence of some action in an image.

Training an actionness detector from videos is a daunting task, as it requires a large number of spatio-temporal annotated frames of actions. In that case, generic action detector is practically useless since one would expect that training generic action detector should be easy as compared to training specific action detector. Fortunately, recent works [29, 70] have demonstrated that deep network trained on images can also provide good classification and localization results in videos. Since that obtaining the bounding box annotations of images is less expensive than for videos, we employ images for the actionness detector. It is still a cumbersome task to get enough bounding box image annotations for a deep network. Therefore, we leverage the internet to obtain relevant images. However, a simple search for "human" and "person" does not work well since it results in producing images contain faces or simple standing people. In contrast, by searching action terms ("man walking", "cricket bowling", "swing", etc.), top retrieved images contain good quality, well-

centered action images that capture the key poses and distinct articulation of actions. Therefore, these images can be a good source for training an actionness detector.

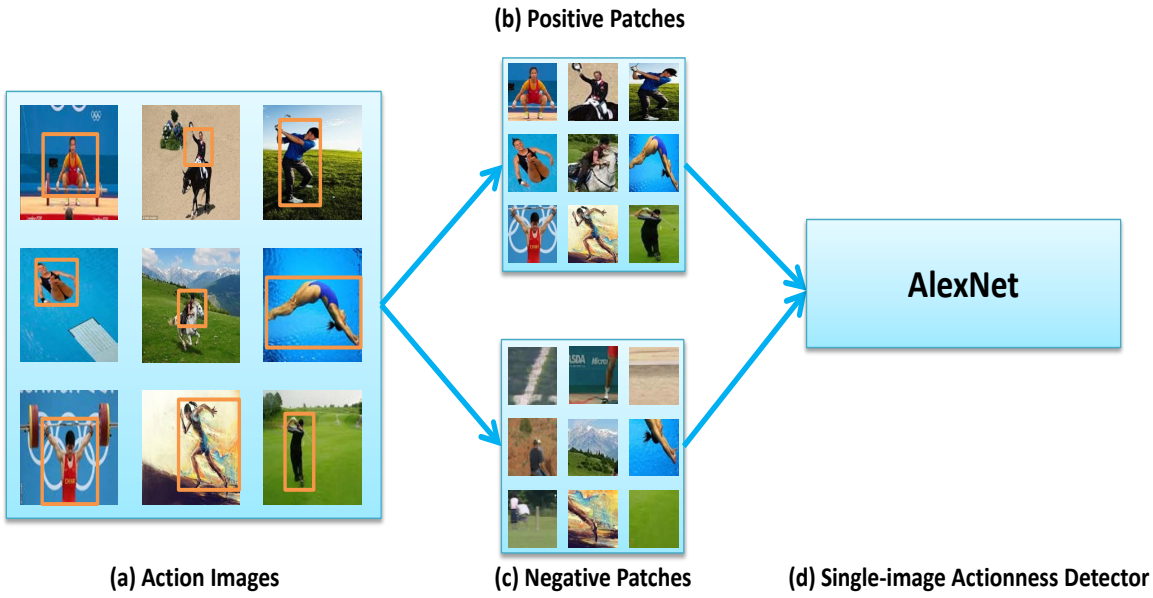

Figure 6.2: Single-image actionness detection. (a) shows some action images downloaded from Google. (b) shows the positive patches obtained by unsupervised bounding box generation. (c) shows some of the low optical flow negative patches obtained from UCF-Sports videos. (d) is the AlexNet based actionness detector.

Positive Images for Image Actionness Detector: We use UCF-Sports (10 actions) [54] action names to download action images from Google. To avoid ambiguity in a text query, we changed some action names slightly, for example using, 'soccer kicking' instead of 'kicking', 'men walking' instead of 'walk'. In order to cope with noisy images and backgrounds within images, we use following pre-processing steps [70].

Noisy Image Removal: Since most of the retrieved images belong to the same concept (the textual query), they form a dense cluster in a feature space [7]. On the other hand, outliers are usually far away from dense clusters or make small clusters. To remove these outliers, we employed random

walk, similar to Chapter 5. We represent each image with 16-layer VGG features, ψ, [83] and make a fully connected graph between all the images. The transition probability of random walk on this graph from image i to image j is given by

$$p(i,j) = \frac{e^{-\alpha\|\psi(i)-\psi(j)\|_2}}{\sum_{m=1}^{k} e^{-\alpha\|\psi(i)-\psi(m)\|_2}}. \tag{6.5}$$

The random walk over the graph is then formulated as:

$$s_k(j) = \gamma \sum_i s_{k-1}(i)p_{ij} + (1-\beta)z_j, \tag{6.6}$$

where s_k represents similarity score of image j at k^{th} iteration. We use the same initial probability z_j for each image. After a fixed number of iterations, we remove the images that receive lower confidence.

Unsupervised Bounding Boxes Generation: Action images that contain actors with significant background may hurt the performance of the classifier. In order to get rid of the background and capture actual action (the common pattern across images), we employ an unsupervised localization approach similar to [8]. First, for all images of the same action, the nearest neighbors are computed using GIST features and then part based matching is performed by employing Probabilistic Hough Matching (PHM) in HOG feature space. The final localization is the patches in the images that contain common pattern across images (action in action images). In order to further remove noisy patches, we repeat random walk framework (described above) on image patches and remove low consistency noisy patches. A few qualitative examples of unsupervised bounding box extraction for these images are shown in Fig. 6.3.

Negative Samples for Actionness Detection: Given the video action proposals in each video, we find the ones that have very low optical flow derivatives and hence probably do not contain

an action. We sample patches from those proposals and use them as negative data for training actionness detector. In practice, we found these patches are more useful compared to non-action images from Google.

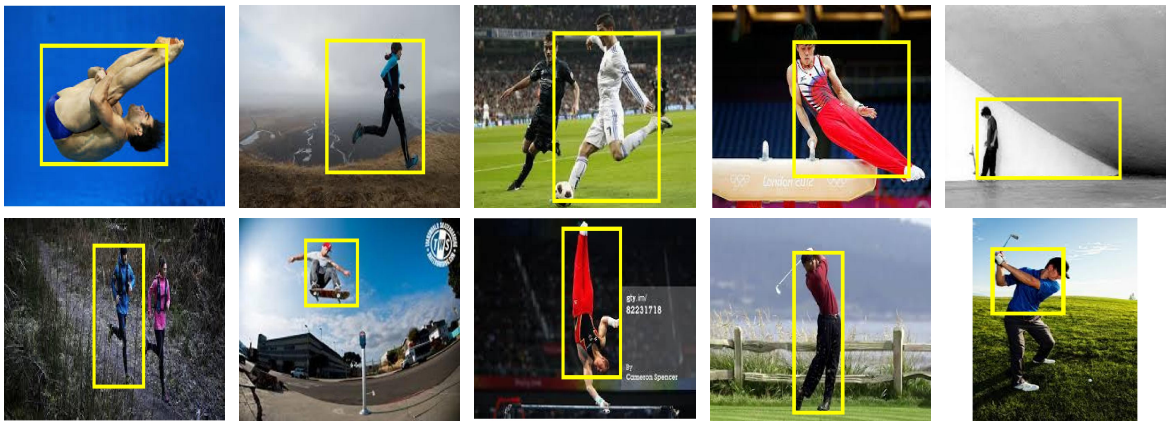

Figure 6.3: Automatically obtained bounding boxes using multiple images. We use these automatically obtained bounding boxes to train our actionness detector. Last column (right) shows the failure cases, where box is either too big (top) or covers only half of the person (bottom).

Image Actionness Detector: Given the positive and negative samples(see Fig. 6.2), we fine-tune AlexNet [36] to obtain actionness detector. We use around 6,000 positive and negative samples for training and 3000 positive and negative samples for validation. For training, the batch size is 256, the learning rate is 10^{-3} and number of iterations is $50K$. The output of this classifier is the probability of each patch being representative of an action. Finally, the actionness score of sub-proposal is the mean of the actionness score of each individual patch within sub-proposal.

6.2.1.2 Motion Score

Actionness measure generic notion of actionnness within the bounding box of each sub-proposal but does not capture the explicit motion information. Due to large camera motion and background

clutter, the straightforward use of optical flow magnitude within each proposal is noisy. Moreover, mostly, a large motion is generated by legs or hands and therefore, the proposals which capture these parts would get high motion scores. However, we want to assign high motion score to the proposals that enclose the complete moving body. Fortunately, motion edges produce high gradients across the whole moving body. We build upon the observation from [101], that higher the number of motion contours enclosed by proposal patch, the higher the probability the patch contains a complete moving body.

Specifically, given the optical flow of two video frames (see Fig 6.4), we compute motion edges for each channel of optical flow separately, The final motion edges are obtained as: E_m=E_u + E_v, where E_u and E_v represent motion edges of *u* and *v* respectively. We, then, use an efficient framework of [101] to obtain several bounding boxes which represent the moving bodies in each frame. Figure 6.4(c) shows motion edges for shown video frames and Figure 6.4(d) shows bounding boxes obtained using these motion edges. In Figure 6.4(e), we show that the bounding boxes obtained using motion edges enclose the action better than the bounding boxes obtained using image edges.

To obtain motion score for any video proposal patch in a particular frame, we compute its overlap with all bounding boxes (obtained using motion edges) in that frame and assign it the score of the highest overlapped bounding box. Finally, the motion score of any sub-proposal is the mean of the motion score of its proposal patches.

6.2.2 Edge Score

The combined actionness and motion scores give the probability that a sub-proposal contains some action; however, it ignores the consistency and similarity between sub-proposal across different proposals. These similarities and consistencies are encoded in the edge score, which is a combination of sub-proposal appearance and shape similarities.

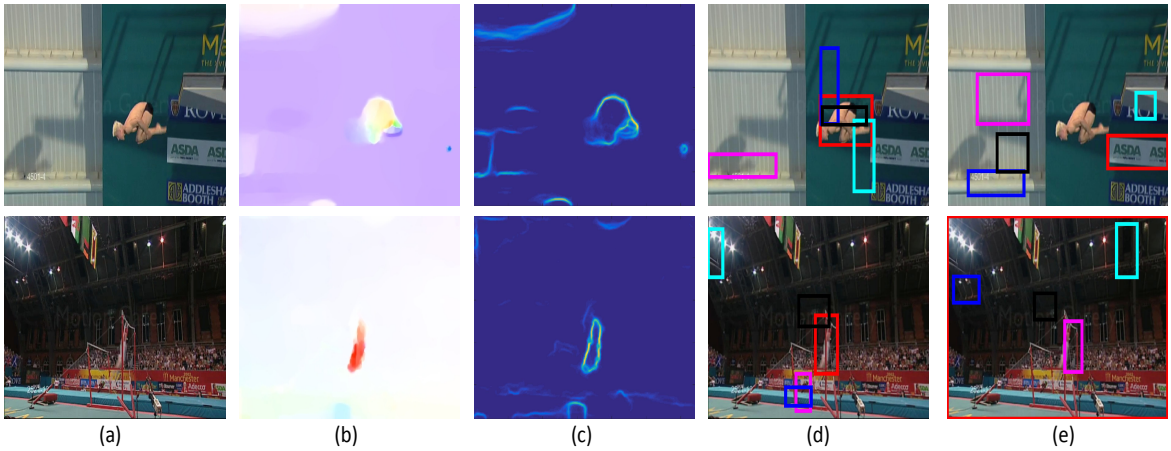

Figure 6.4: (a) video frames, (b) optical flow, (c) motion edges, (d) proposals computed based on motion edges and (e) proposals computed based on image edges. The red bounding box shows the highest scored proposal.

The shape similarity of two sub-proposal is measured by intersection-over-union (IOU) between last frame in one sub-proposal and the first frame of its temporally next sub-proposal. It is defined as:

$$\Psi^o = \frac{Area(b_{i,l} \cap b_{j,1})}{Area(b_{i,l} \cup b_{j,1})}, \tag{6.7}$$

where $b_{i,l}$ represents bounding box in the last frame of a sub-proposal i and $b_{j,1}$ represents the bounding box in the first frame of its temporally next sub-proposal j. To capture appearance similarity Ψ^a, we use Euclidean similarity between mean of HOGs of patches within each sub-proposal.

6.3 Experiments

The main goal of our experiments is to evaluate the accuracy of proposal ranking for trimmed and untrimmed datasets and validate proposal ranking effectiveness for action detection. In all the

experiments, we use λ=λ^i=λ^m=λ^o=λ^a = 1 (in Equation (6.1) and (6.2)).

6.3.1 Proposal Ranking in Trimmed Videos

We evaluated the proposed approach on two challenging datasets: UCF-Sports [54] and sub-JHMDB [40]. In order to calculate localization accuracy, we compute overlap of the top ranked action proposal with ground truth annotations as defined in [27]. Since our approach is generic as it does not depend on any specific underlying action proposal method, we demonstrate localization results for three recent action proposals methods [49, 27, 81]. The method proposed in [49, 27] are based on supervoxel segmentation and hierarchical merging of those segments. While van Gemert et al. [81] used improved dense trajectories [86] clustering to obtain action proposals. We first compute optical flow derivatives for each proposal and remove highly overlapped proposals using non-maximal suppression.

Given that our method does not need any training, we report results on all videos in the datasets. We compared our approach with two recently proposed *weakly supervised* proposal ranking methods [75, 63]. These methods achieve proposals ranking by exploiting proposals from negative videos, i.e., by using proposals from the videos that contain actions other than the action of interest. Specifically, [63] ranked the proposals according to their nearest neighbor in negative videos. Larger the distance of proposal to the nearest neighbor proposal in the negative video, the higher is the rank of the proposal. The method in [75] improved over [63] and employed all negatives proposals and penalize proposals to be ranked using their distance to negative proposals. We implemented both methods as described in [75]. For proposals representation, we use CNN features [83] within proposal patch averaging over sample frames in the proposal, similar to [29].

In Table 6.2, we demonstrate localization accuracy of top most proposal over different thresholds. We compared our approach with both methods [75, 63] for all proposals methods we use.

Figure 6.5: Qualitative results for UCF-Sports. Each row shows four frames of videos from two UCF-Sports actions. The magenta and green boxes respectively indicate ground truth and top ranked action proposal.

We do not use Jain et al. [27] proposals for Sub-JHMDB, due unavailability of their code for this dataset.

It can be seen in Table 6.2 that for both datasets and for all proposal methods, our method significantly outperforms two baselines without even using video level labels. As we increase the threshold, the localization accuracy decreases; however, our results are always better compared to the baselines. At overlap threshold of 20%, as used by several recent works for action detection [76, 66, 27], our method has more than 85% accuracy for both datasets. In Figure 6.7, we demonstrate recall versus different number of proposals. The results demonstrate that our approach significantly outperforms baselines for all different number of proposals.

Figure 6.6: Qualitative examples from Sub-JHMDB. Each block shows four frames from a different action video in Sub-JHMDB. The magenta and green boxes respectively indicate ground truth and top ranked action proposals.

Fig. 6.5 and Fig. 6.6 show the qualitative results for top-ranked proposals for different videos of UCF-Sports and Sub-JHMDB. It can be seen that top ranked proposals cover the actor very well despite the presence of many noisy proposals due to clutter, large camera motion, and dynamic backgrounds.

Table 6.1. shows the contribution of actionness score and motion score for final action localization. The top row shows that using Motion score and Shape + Appearance similarities (for graph construction), we achieve 40.4 (MABO). Second-row shows using Actionness and Shape + Ap-

pearance gives 28.9. The last row demonstrates that motion and actionness have complementary information and combination of both results in total MABO of 45.5. Finally, third and fourth rows show localization accuracies using individual graph edge component. The results demonstrate that each component of our method is essential to achieve final localization accuracy.

It is interesting to observe that through proposal recombination, in many videos, we are able to discover proposals that are better than the best existing proposal in the initial proposal method. Given top 20 proposals, for UCF Sports and Sub-JHMDB, in 64 and 83 videos respectively, our newly generated proposals are better than all underlying proposals [49].

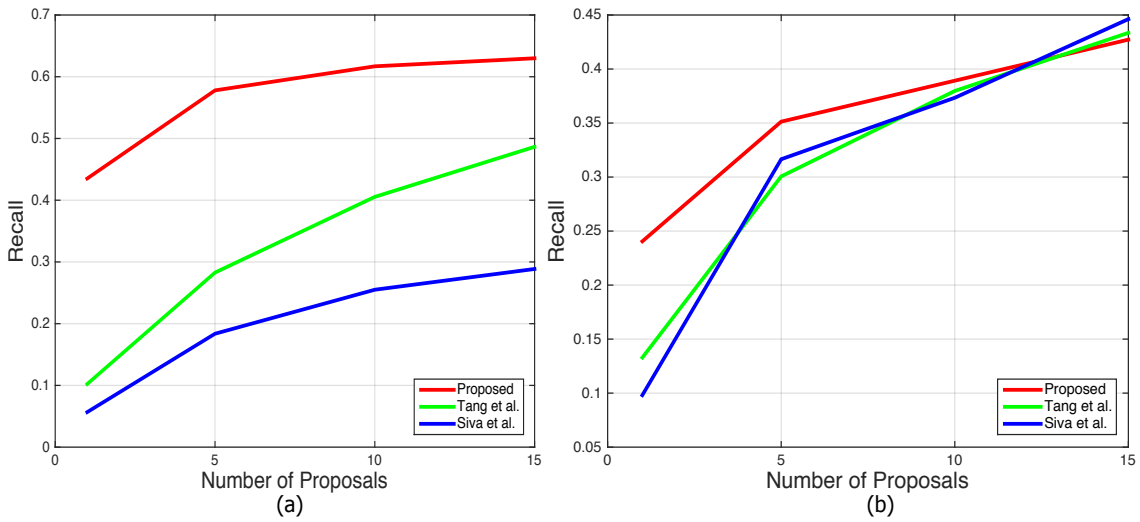

Figure 6.7: Recall versus number of proposals for UCF-Sports (a) and sub-JHMDB (b). The results are shown for Proposed method (red), Tang et. al. (green) and Siva et. al. (blue).

6.3.2 *Proposal Ranking in Untrimmed Videos*

In addition to trimmed datasets, we tested our approach on untrimmed **MSR-II** dataset. This dataset has 54 untrimmed videos and contains three actions: Handwaving, Clapping, and Boxing. This dataset is very challenging due to background clutter and occlusions.

Since these videos are untrimmed, we first employ an unsupervised shot-detection method [2] to divide the long video sequences to small shots and limit the action proposals within the shots. Below, we describe in details shot detection procedure.

Temporal Segmentation: We compute the mean location of the action proposals in each video frame, and then the action proposal speed is computed by the location distance between adjacent frames. For action proposal diversity, we calculate the standard deviation of the locations of action proposals in every video frame. Finally, for the frame difference, we simply calculate the sum of the pixel differences between adjacent video frames. Given these three features, we employ a thoroughly studied and well developed multiple change point detection scheme technique in statistic [2] for detecting shots. Let $\mathbf{Z} \in \mathbb{R}^{N \times K}$ represents the matrix obtained by horizontal concatenation of features vectors of length K from N video frames. Note that in our case, since we have 3 features, K=3.

Table 6.1: Contribution from different components on UCF-sports with van Germert et al. proposals. In the following table, M corresponds motion score, Ap, and S represents appearance and shape similarity respectively and Ac represents actionness score

Components	**t=0.1**	**t=0.2**	**t=0.3**	**t=0.4**	**t=0.5**	**t=0.6**	**MABO**
M + S + Ap	94.2	88.3	74.0	52.6	31.2	11.4	40.4
Ac + S + Ap	81.2	65.6	45.4	23.4	14.9	7.8	28.9
Ac + M + Ap	96.1	92.9	83.8	64.9	47.4	16.9	44.9
Ac + M + S	96.1	92.7	83.8	62.9	39.6	16.9	44.9
M + Ac + S + Ap	94.8	92.9	83.7	64.3	43.5	18.2	45.5

Table 6.2: Comparisons of Proposals Ranking using Different Action Proposal

Datasets	Proposal	Methods	t=0.2	t=0.5	MABO
UCF-Sports	Jain et al.	**Ours**	**90.3**	**39.6**	**43.9**
		Tang et.al.	50.4	15.7	25.1
		Siva et.al.	48.9	12.0	23.3
		Jain et.al.	22.6	8.8	12.2
	Oneata et al.	**Ours**	**89.6**	**44.2**	**44.7**
		Tang et.al.	47.9	8.4	24.3
		Siva et.al.	61.4	11.0	27.5
		Oneata et al.	45.9	7.8	21.2
	van Gemert et al.	**Ours**	**92.9**	**43.5**	**45.5**
		Tang et.al.	47.4	10.2	23.6
		Siva et.al.	51.1	5.7	23.3
		van Gemert et al.	40.6	6.7	19.0
Sub-JHMDB	Oneata et al.	**Ours**	**84.5**	**38.6**	**42.9**
		Tang et.al.	39.6	13.3	22.2
		Siva et.al.	61.7	9.8	26.9
		Oneata et al.	36.7	5.2	18.2
	van Gemert et al.	**Ours**	**90.5**	**24.1**	**39.5**
		Tang et.al.	71.2	2.5	25.9
		Siva et.al.	74.0	7.6	28.8
		van Gemert et al.	58.1	6.6	25.1

We seek to identify piece-wise approximation of $\mathbf{Z}$, i.e., $\mathbf{X}$ using following convex optimization formulation:

$$\min_{\mathbf{X}} \frac{1}{2} \|\mathbf{Z} - \mathbf{X}\|^2 + \lambda \sum_{n=1}^{N-1} \frac{\|X_{n+1,.} - X_{n,.}\|}{w_n}, \tag{6.8}$$

where number of shots are automatically determined by convex total variation [2] and w_n represents temporal location-dependent weights given as:

$$w_n = \sqrt{\frac{N}{n(N-n)}}. \tag{6.9}$$

Note that the first term in Equation. 1 represents quadratic error criterion and the second term enforce row-wise sparsity in X. We solve above equation efficiently using group LARS [97].

On average, we have 9 shots (unsupervisedly selected) in each video and we found top ranked proposals in each shot. We compared our method with recently proposed state-of-the-art supervised action proposals method [96] and report CorLoc results in Figure. 6.8. It can be seen that even with 30% less number of proposals, our *unsupervised* approach performs better than the *supervised* action proposal method. Furthermore, we show the results of [81] (blue curve) with the number of proposals equal to our method. The quantitative results shown in Figure 6.8 demonstrates that our method significantly outperforms when considering less number of proposals.

In the next section, we discuss the action detection using re-ranked proposals.

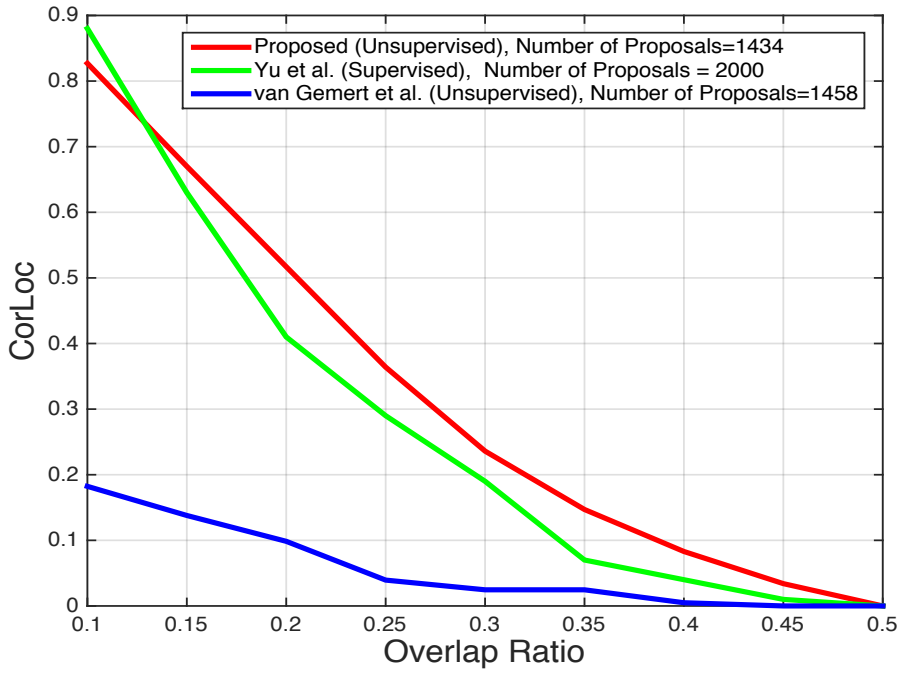

Figure 6.8: CorLoc comparision for different thresholds on MSR-II.

6.3.3 Action Detection

Better-ranked action proposals not only improve computation efficiency by reducing the search space but also improve action recognition accuracy by reducing the false positive rate. We closely follow the procedure described in [81] for proposal classification. Specifically, we concatenate dense trajectory features (MBH, HOG, HOF, Traj) into a 426-dimensional vector and reduce its dimension to half using PCA. We randomly sample features from training videos and fit 128 GMMs to it. We compute fisher vector representation of each proposal followed by power and L_2 normalization. Finally, we use a linear classifier for training action detector. In our case, during testing, we only use top 25 proposals for features computation as well as classification. The final prediction scores of top 25 proposals are obtained by multiplying proposal scores and classifier scores. In Fig. 6.9, we compare our results with two recent state-of-the-art proposal based action detection methods. It can be seen that we outperform both methods with significant margin at all overlapping thresholds.

Similar to UCF-Sports, we perform action detection experiments on UCF101. We use [81] for getting initial proposals. Experimental results in Figure 6.9 shows the superiority of our approach as compared to baseline [81]. Note that author in [81] have used 2299 proposals in each video. In contrast, our method just use only 200 proposals in each video.

Performing better than the baselines emphasizes the strength of the proposed approach and reinforces that properly ranked action proposals have significant impact on action detection.

6.3.4 Computation Time:

Our approach works on top of [7][4][5]. These proposals take few minutes for computing proposal in each video. We now have optimized our code slightly. Our code takes 0.002 seconds to compute

action score for each proposal patch and 0.02 seconds for optimal path discovery for both UCF-Sports and sub-JHMDB datasets. Other steps: optical flow derivatives, HOGs similarity also take less than a sec.

6.4 Summary

In this chapter, we present a method for action proposal ranking which aims to generate fewer but better action proposals which are ranked properly. We test our method on UCF-Sports, sub-JHMDB and MSR-II for action proposal ranking and compared it to several baseline methods. We show that the proposed method generates fewer but better proposals. The method is useful for many applications and can reduce the search space dramatically.

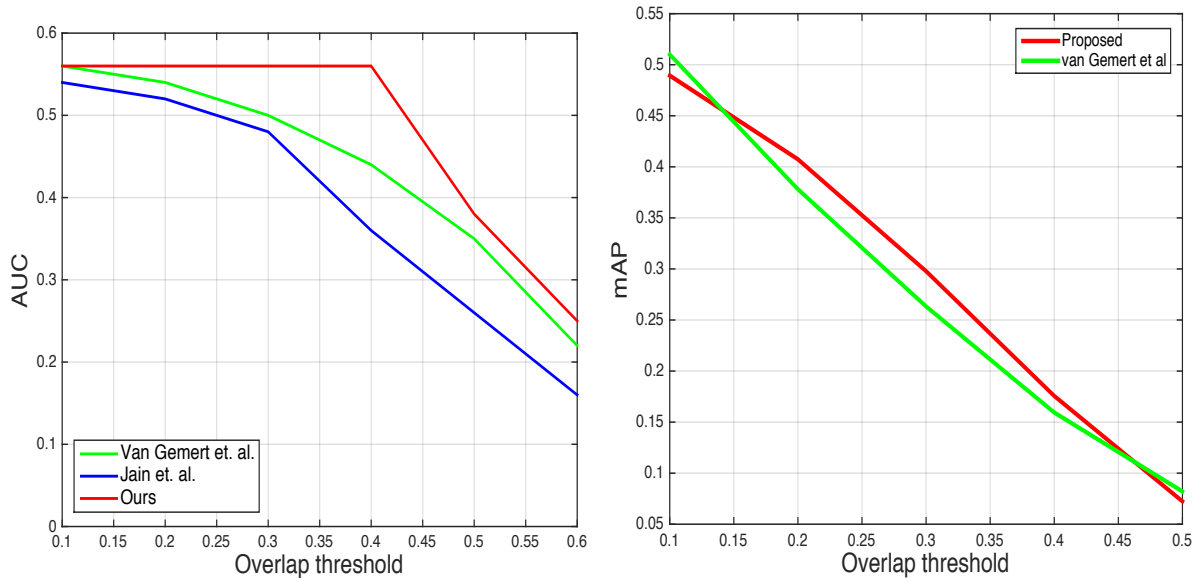

Figure 6.9: Left: The AUC curves for UCF-Sports dataset [54]. The results are shown for proposed method (red), Jain et al. [27] (blue) and Van Gemert et al. [81] (green). Right: mAP curves on UCF101 datasets are shown for Proposed method (red) and van Van Gemert et al. [81] (green)

We also perform experiments for action detection and showed better performance compared with state-of-the-art proposals based action detection methods.

CHAPTER 7: CONCLUSION AND FUTURE WORK

7.1 Conclusion

In this dissertation, we address the problem of action recognition and localization in weakly labeled videos. We discuss that action datasets have discriminative backgrounds, which may affect the generalizability of action classifiers. We, then, present a novel approach for automatic annotation of an action by exploiting similarity across multiple videos. Moreover, we demonstrate that it is possible to obtain action localization in a single video by leveraging web images searched through text queries. Finally, we present a new approach to obtain a few properly ranked action proposals through proposal recombination.

In chapter 3, we experimentally demonstrate the detrimental effect of background scenes in action recognition dataset. Using motion and scene features, we show that backgrounds in recent complex datasets do have discriminative effects, which artificially inflate action recognition accuracy. In order to mitigate the effect of backgrounds and improve the generalizability of action classifiers, we propose a new process for obtaining per pixel confidence of every video pixel being the foreground, as well as novel soft assignment, and histogram decomposition schemes for the bag-of-words representation. We have demonstrated the use of foreground focused action representation in cross-dataset action recognition where neither the labels nor the features of test set are available.

Building on some of the ideas discussed in Chapter 3, we address an important problem of automatic annotations in Chapter 4. Manual spatio-temporal annotation of human actions in videos requires considerable effort and time. By estimating foreground confidences of action proposals in each video, we get rid of background proposals. After that, we use similarities across multiple videos to capture action localization in each video. In contrast to expensive and laborious

annotations, we obtain these spatio-temporal annotation boxes automatically by matching action proposals across multiple videos using their feature and shape similarities. Moreover, we demonstrate that these annotations can be used to learn robust action classifiers.

In order to address limitations of the approach discussed in Chapter 4, we present a new approach to spatio-temporally localizes an action in a single video. To address noise, memory and computational limitations of previous approaches, we use web images without assuming the availability of multiple videos, prior annotations or clean images. We demonstrate the feasibility of web images for action localization. For this purpose, we use sparse reconstruction along consensus regularization techniques to reconstruct video frames using web images and use reconstruction error as a measure of similarity between web images and video frames.

Finally, to improve current action proposal methods, we propose to obtain a few properly ranked proposals through proposal combination. We propose a new actionness and motion score for each sub-proposals and then use Dynamic Programming based graph optimization scheme to select the optimal combinations of sub-proposals from different proposals and assign each new proposal a score. We experimentally demonstrate that our approach is generic and does not depend on a specific action proposal method.

We conduct extensive experimental on different challenging and realistic action datasets and compared our approaches with several competitive baselines to validate the effectiveness of proposed ideas and frameworks.

7.2 Future Work

There are several possible directions of the future works that can improve the accuracy and efficiency of approaches proposed in this dissertation, as discussed below.

Foreground focused representation, presented in this dissertation, depends on low-level unsupervised saliency estimation and optical flow derivatives. However, the use of supervised saliency and foreground motion estimation is expected to give much better probabilities of foreground regions. Moreover, we have used Bag of Words to validate foreground focused representation within and across datasets. However, Bag of Words itself has much less accuracy as compared to recent deep learning frameworks. Fusion of deep learning and foreground representations can significantly boost the accuracy of within and across datasets experiments. Moreover, training action classifiers on a large number of videos, either from multiple datasets or from the web, can learn more generalized action model and would perform better across different datasets.

Automatic action annotation framework can also be improved in several ways. For action proposals matching, we mainly employ improved dense trajectory features and Bag of Words representation. Proposed global similarity measure can be improved by replacing the Bag of Words representation with 3D convolution features [78]. We have used three cues for proposal matching: global, fine-grained and shape. It would be worth to investigate more cues for proposal matching. Furthermore, in order to find the most consistent proposal in each video, we have used GMCP. However, the GMCP solver that we have used in our experiments, does not guarantee an optimal solution and has a potential to stuck in local maxima. A better solver could find a better solution and have the potential to get better annotations. GMCP selects only one maximum clique at a time. A solver that can find all maximum cliques jointly will be more useful. Finally, for larger datasets such THUMOS13, GMCP optimization is quite slow. Therefore, use of faster graph optimization technique would improve the efficiency of the proposed approach.

Action localization using Web images is an interesting work, however, there are several components that need more detailed analysis. Our results show that removing noisy images can improve localization accuracy. Instead of using random walks, better noise removal approach is expected to bring more improvements. Although, our initial experimental results using videos and images

jointly instead of images only does not improve accuracy for action localization task; a more detailed investigation may bring some new insights. Since images capture static instances of an action, they cannot discriminate very well between action and non-action frames in untrimmed action videos. Furthermore, traditional shot detection method, that we employ in our work, sometimes fail to precisely detect temporal action boundaries. More research in this direction can bring significant improvements in action localization in untrimmed videos.

In order to train actionness detector for action proposal ranking, we fine-tune AlexNet using web images. Replacing AlexNet with the recent more deep network is expected to produce more robust detector. We discover multiple top ranked proposals one by one using dynamic programming based solution. However, graph optimization technique that can discover all multiple top ranked proposals at the same time can boost the efficiency of our approach

The proposed approaches assume video level labels for training action detector, however, it would be an interesting direction to extend the ideas discussed in this dissertation to completely unsupervised setting i.e., where the video level labels are also unknown. Furthermore, recognition and localization of human interactions that involves multiple people are valuable for real-world practical applications. Moreover, the method discussed in this thesis are off-line and assume the availability of a complete video for action localization. However, online versions of the proposed methods are more useful for real-time action localization applications.

LIST OF REFERENCES

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010.

[2] K. Bleakley and J.-P. Vert. The group fused lasso for multiple change-point detection. In *arXiv preprint arXiv:1106.4199*, 2011.

[3] H. Boyraz, S. Z. Masood, B. Liu, M. Tappen, and H. Faroosh. Action recognition by weakly-supervised discriminative region localization. In *BMVC*, 2014.

[4] L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *PAMI*, 32(5):770–787, 2010.

[5] C.-Y. Chen and K. Grauman. Watching unlabeled video helps learn new human actions from very few labeled snapshots. In *CVPR*, 2013.

[6] L. Chen, L. Duan, and D. Xu. Event recognition in videos by learning from heterogeneous web sources. In *CVPR*, 2013.

[7] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013.

[8] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, 2015.

[9] A. Dehghan, H. Idrees, and M. Shah. Improving semantic concept detection through the dictionary of visually-distinct elements. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[10] A. B. Deselaers, T. and V. Ferrari. Weakly supervised localization and learning with generic knowledge. In *IJCV*, 2012.

[11] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *PETS*, 2005.

[12] L. Duan, D. Xu, and I. W.Tsang. Domain adaptation from multiple sources: A domain-dependent regularization approach. 2012.

[13] L. Duan, D. Xu, I. W.Tsang, and J. Luo. Visual event recognition in videos by learning from web data. In *CVPR*, 2010.

[14] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015.

[15] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *CVPR*, 2004.

[16] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[17] C. Feremans, M. Labbe, and G. Laporte. Generalized network design problems. *EJOR*, 2003.

[18] H. Fu, D. Xu, B. Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *CVPR*, 2014.

[19] E. Gafni and D. Bertsekas. Two-metric projection methods for constrained optimization. In *SIAM Journal on Control and Optimization*, 1982.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[21] G. Gkioxari and J. Malik. Finding action tubes. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, June 2015.

[22] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[23] M. Guillaumin and V. Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *CVPR*, 2012.

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

[25] http://crcv.ucf.edu/ICCV13 Action-Workshop/. Thumos: The first international workshop on action recognition with a large number of classes. 2013.

[26] H. D. III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[27] M. Jain, J. van Gemert, H. Jegou, P. Bouthemy, and C. Snoek. Action localization with tubelets from motion. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 740–747, June 2014.

[28] M. Jain, J. C. van Gemert, and C. G. M. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[29] M. Jain, J. C. van Gemert, and C. G. M. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, June 2015.

[30] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, Dec. 2013.

[31] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`.

[32] A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *ECCV*, 2014.

[33] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, 2013.

[34] G. Kim, L. Sigal, and E. Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014.

[35] A. Kläser, M. Marszałek, I. Laptev, and C. Schmid. Will person detection help bag-of-features action recognition? Technical Report RR-7373, INRIA Grenoble - Rhône-Alpes, 2010.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*. 2012.

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

[38] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[39] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[40] Y. Q. L. Wang and X. Tang. Video action detection with relational dynamic-poselets. 2014.

[41] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.

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

[43] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

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

[45] S. Manén, M. Guillaumin, and L. Van Gool. Prime Object Proposals with Randomized Prim's Algorithm. In *ICCV*, Dec. 2013.

[46] S. Manen, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized prim's algorithm. In *ICCV*, 2013.

[47] H. Moonesinghe and P.-N. Tan. Outlier detection using random walks. In *Tools with Artificial Intelligence, 2006. ICTAI '06. 18th IEEE International Conference on*, 2006.

[48] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42:145–175, 2001.

[49] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid. Spatio-Temporal Object Detection Proposals. In *ECCV*, 2014.

[50] D. P. Papadopoulos, A. D. F. Clarke, F. Keller, and V. Ferrari. Training object class detectors from eye tracking data. In *ECCV*, 2014.

[51] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[52] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *MVA*, 24(5):971–981, 2012.

[53] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[54] M. Rodriguez, J. Ahmed, and M. Shah. Action MACH: A spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[55] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[56] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[57] M. Schmidt. Graphical model structure learning with l1-regularization. In *Ph.D. Thesis*, 2010.

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

[59] N. Shapovalova, A. Vahdat, K. Cannons, T. Lan, and G. Mori. Similarity constrained latent support vector machine: An application to weakly supervised action classification. In *ECCV*, 2012.

[60] E. Shechtman, L. Gorelick, M. Blank, M. Irani, and R. Basri. Actions as space-time shapes. *PAMI*, 29(12):2247–2253, 2007.

[61] R. Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. In *The Computer Journal*, 1973.

[62] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27*, 2014.

[63] P. Siva, C. Russell, and T. Xiang. In defence of negative mining for annotating weakly labelled data. In *ECCV*, 2012.

[64] P. Siva and T. Xiang. Weakly supervised action detection. In *BMVC*, 2011.

[65] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, 2015.

[66] K. Soomor, H. Idrees, and M. Shah. Action localization in videos through context walk. In *ICCV*, December 2015.

[67] K. Soomro, R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *ICCV*, 2013.

[68] Y. Su and F. Jurie. Visual word disambiguation by semantic contexts. In *ICCV*, 2011.

[69] W. Sultani and I. Saleemi. Human action recognition across datasets by foreground-weighted histogram decomposition. In *CVPR*, June 2014.

[70] W. Sultani and M. Shah. What if we do not have multiple videos of the same action? – video action localization using web images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[71] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *Proceedings of the 23rd ACM International Conference on Multimedia*, 2015.

[72] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, pages 1464–1471, June 2014.

[73] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014.

[74] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller. Shifting weights: Adapting object detectors from image to video. In *NIPS*, 2012.

[75] K. Tang, R. Sukthankar, J. Yagnik, and L. Fei-Fei. Discriminative segment annotation in weakly labeled video. In *CVPR*, 2013.

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

[77] A. Torralba and A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

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

[79] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. In *IJCV*, 2013.

[80] M. M. Ullah, S. N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *BMVC*, 2010.

[81] J. van Gemert, M. Jain, G. Ella, and C. Snoek. Apt: Action localization proposals from dense trajectories. In *BMVC*, pages 740–747, June 2015.

[82] J. Van Gemert, C. Veenman, A. Smeulders, and J.-M. Geusebroek. Visual word ambiguity. *PAMI*, 32(7):1271–1283, 2010.

[83] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. 2015.

[84] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[85] H. Wang and C. Schmid. Action recognition by dense trajectories. In *ICCV*, 2013.

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

[87] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[88] X. Wang, A. Gupta, undefined, undefined, undefined, and undefined. Unsupervised learning of visual representations using videos. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[89] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[90] G. Willems, T. Tuytelaars, and L. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.

[91] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012.

[92] J. Yang, R. Yan, and A. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ICM*, 2007.

[93] M. Yang, L. Zhang, D. Zhang, and S. Wang. Relaxed collaborative representation for pattern classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012.

[94] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011.

[95] Z. Yin and R. Collins. Belief propagation in a 3d spatio-temporal mrf for moving object detection. In *CVPR*, 2007.

[96] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, June 2015.

[97] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. 2006.

[98] A. Zamir and M. Shah. Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. *PAMI*, 2014.

[99] D. Zhang, O. Javed, and M. Shah. Video object co-segmentation by regulated maximum weight cliques. In *ECCV*, 2014.

[100] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Měch. Unconstrained salient object detection via proposal subset optimization. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2016.

[101] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.