The Thesis Committee for Sunil Bandla

certifies that this is the approved version of the following thesis:

Active Learning of an Action Detector on Untrimmed Videos

APPROVED BY

SUPERVISING COMMITTEE:

Kristen Grauman, Supervisor

Raymond Mooney

Active Learning of an Action Detector on Untrimmed Videos

by

Sunil Bandla, B. Tech

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Computer Science

The University of Texas at Austin

May 2013

Acknowledgments

I am fortunate to have worked with Professor Kristen Grauman for my Masters thesis. This work would not have been possible without her guidance and patience. Her work ethic and planning are truly inspiring. Words cannot convey my gratitude to her for believing in me, and supporting me.

I would like to thank Professor Raymond Mooney for his valuable comments on this thesis.

I would also like to thank my labmates for all their help and advice in various matters. The random discussions we had over drinks and the more recent table tennis games were fun and memorable.

Finally, my heartfelt appreciation to my friends and family for their encouragement, love and support throughout my stay here. This work exists because of the persistence of my iron-willed mother Sasikala Bandla and my amazingly helpful brother Sudheer Bandla. I dedicate this thesis to my late father Prasada Rao Bandla.

SUNIL BANDLA

The University of Texas at Austin

May 2013

Active Learning of an Action Detector on Untrimmed Videos

Sunil Bandla, M.S.Comp.Sci.
The University of Texas at Austin, 2013

Supervisor: Kristen Grauman

Collecting and annotating videos of realistic human actions is tedious, yet critical for training action recognition systems. We propose a method to actively request the most useful video annotations among a large set of unlabeled videos. Predicting the utility of annotating unlabeled video is not trivial, since any given clip may contain multiple actions of interest, and it need not be trimmed to temporal regions of interest. To deal with this problem, we propose a detection-based active learner to train action category models. We develop a voting-based framework to localize likely intervals of interest in an unlabeled clip, and use them to estimate the total reduction in uncertainty that annotating that clip would yield. On three datasets, we show our approach can learn accurate action detectors more efficiently than alternative active learning strategies that fail to accommodate the "untrimmed" nature of real video data.

Contents

Chapter 1

Introduction

Locating where an action occurs in a video is known as *action localization* or *action detection*, while categorizing an action is called *action recognition*. The difficulty of these tasks arises from several factors, such as intra-class variations, pose changes, incomplete actions, and clutter. Researchers have made much progress in recent years to deal with these issues, most notably with learning-based methods that discover discriminative patterns to distinguish each action of interest, e.g. [28, 14, 16, 6, 43, 3, 22].

Of course, good learning requires good data. Unfortunately, data collection is particularly intensive for activity recognition, since annotators must not only identify what actions are present, but also specify the time interval (and possibly spatial bounding box) where that action occurs. With insufficient data, methods are apt to overfit and will fail to generalize to intra-class variations at test time. Yet, the amount of manual supervision required to annotate large datasets [12, 23] with many action categories can be daunting and expensive.

While *active learning* is a natural way to try and reduce annotator effort, its success has been concentrated in the object recognition literature (e.g., [5, 32, 30, 33]). In fact, to our knowledge, the only active learning efforts for video have focused on problems other than activity recognition—namely, segmenting tracked objects [36, 7, 34] or identifying

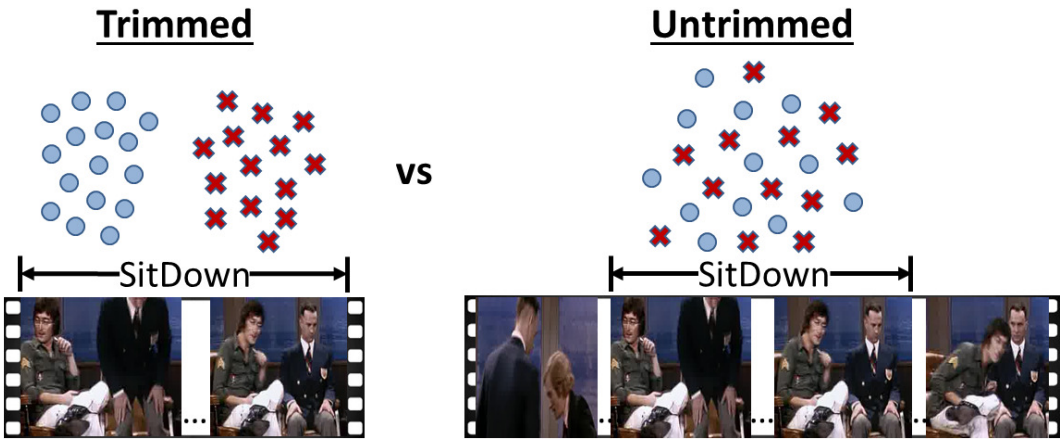


Figure 1.1: **Left:** If we somehow had unlabeled videos that were trimmed to the action instances of interest (just not labeled by their category), then useful instances would be relatively apparent, and one could apply traditional active learning methods directly. **Right:** In reality, though, an unlabeled video is *untrimmed*: it may contain multiple action classes, and it need not be temporally cropped to where actions of interest occur. This results in an unlabeled feature distribution where useful and redundant candidates are hard to distinguish. Our approach takes untrimmed and unlabeled videos as input and repeatedly selects the most useful one for annotation, as determined by its current action detector.

people [40] in video frames. We see that an important challenge in attempting to actively select video for learning actions is video's ongoing, "untrimmed" nature. Before passing through the hands of an annotator, a typical video will *not* be trimmed in the temporal dimension to focus on a single action (let alone cropped in the spatial dimensions). In contrast, a typical image snapshot depicts only a limited number of objects, and may even focus on a primary foreground object thanks to human photographer framing tendencies. That makes images and objects more amenable to standard active learning paradigms.

Applying active learning to video clips is non-trivial. Most active learning algorithms assume that data points in the unlabeled pool have a single label, and that the feature descriptor for an unlabeled point reflects that instance alone. Yet in real unlabeled video, there may be multiple simultaneous actions as well as extraneous frames belonging to no action category of interest. As a result, untrimmed videos cannot be used directly to estimate standard active selection criteria. For example, an uncertainty-based sampling strat-

egy using a classifier is insufficient; even if the action of interest is likely present in some untrimmed clip, the classifier may not realize it if applied to all the features pooled together. Requesting labels on such video would only waste the annotator's effort. Figure 1.1 depicts the underlying problem.

Our goal is to perform active learning of actions in untrimmed videos. To achieve this, we introduce a technique to measure the information content of an untrimmed video, rank all unlabeled videos based on these scores, and request annotations on the most valuable video.

The method works as follows. In order to predict the informativeness of an untrimmed video, we first use an incrementally updated Hough-based action detector to estimate the spatio-temporal extents in which an action of interest could occur. Whereas a naive approach would evaluate all possible spatio-temporal intervals, this step allows us to focus on a small subset of candidates per video. Next, we forecast how each of these predicted action intervals would influence the action detector, were we to get it labeled by a human. To this end, we develop a novel uncertainty metric computed based on entropy in the 3D vote space of a video. Importantly, rather than simply look for the single untrimmed video that has the highest uncertainty, we estimate how much each candidate video will reduce the *total* uncertainty across *all* videos. That is, the best video to get labeled is the one that, once used to augment the Hough detector, will more confidently localize actions in all unlabeled videos. We ask a human to annotate the most promising video, and use the results to update the detector. The whole process repeats for a specific number of rounds, or until a given budget of manual annotation effort is exhausted.

We evaluate our method on three action localization datasets. We examine recognition accuracy as a function of the number of annotated videos, comparing our method to both passive and active alternatives. The results demonstrate that accounting for the untrimmed nature of unlabeled video data is critical; in fact, we find directly applying a standard active learning criterion can even underperform the passive learning strategy. In

contrast, the proposed approach offers substantial savings in human annotation effort, as it identifies the most useful videos to label. Furthermore, our experiments offer insight into the important next steps for research in this area.

Chapter 2

Related Work

To recognize actions, some approaches leverage body tracking and shape analysis [21, 2, 18, 24], while others use the overall appearance and motion patterns in a video clip by detecting local spatio-temporal interest points [13, 39] often followed by visual vocabulary formation and Support Vector Machine (SVM) classification [28, 14, 16].

While most work focuses on the action *recognition* task alone, some recent work tackles action *detection*, which further entails the localization problem. There are three main strategies: tracking-based, sliding window, and voting-based. Person-centric detection approaches localize actions with the help of person tracking [21, 10, 41], while sliding window methods check the classifier's response for all possible subvolumes [6, 27], possibly incorporating efficient search strategies [43, 42, 3]. Compared to the both of the above, voting-based methods for action detection [17, 38, 41] have the appeal of circumventing person tracking (a hard problem of its own), allowing detection to succeed even with partial local evidence, and naturally supporting incremental updates to the training set. Our active learning method capitalizes on these advantages, and we adapt ideas from [38, 41] to build a Hough detector (Sec. 3.2). A voting-based method for object detection [15] treats features detected in an image as object parts and uses them to cast votes in a 2D space for the object center in an unseen image. The object center is then obtained by finding peaks in the vote

space. While we tailor the details to best suit our goals, the novelty of our contribution is the active learning idea for untrimmed video, not the action detector it employs.

Active learning has been explored for object recognition with images [9, 20, 5, 32, 30, 11, 33] and to facilitate annotation of objects in video [40, 36, 7, 34]. Object recognition models are built in [32] using active learning with multiple instance learning (MIL) to identify the type of annotation most useful on an image in the unlabeled set. Contextual relationships between regions in an image are studied to find regions that need to be labeled in an image [30] and active selection of object windows in images to be sent for annotation is tackled in [33]. Among the above mentioned techniques, methods that accommodate unlabeled multi-object images [32, 30, 33] have some parallels to our problem: while our method must reason about the informativeness of any possible space-time region in an untrimmed video, those methods must reason about the informativeness of objects whose exact spatial extents within the image are unknown. However, whereas the image-based methods can exploit powerful segmentation algorithms to identify candidate regions that might be objects, there is no obvious analogy to identify candidate regions that might be individual actions in video; part of our contribution is showing how voting in space-time can generate good candidates. Unlike any of the above methods, we propose to actively learn an action detector.

Active learning itself has been studied for many years in machine learning [29]. As discussed above, untrimmed video makes direct application of existing active learning methods problematic, both conceptually and empirically, as we will see in the results. We devise a selection criterion (Sec. 3.4) based on the classic idea of expected error reduction as evaluated on unlabeled data, which is statistically optimal [4, 25]. Various forms of expected error reduction have been explored in many domains, including object recognition [32, 30, 11]. In [11], active learning is used to determine the type of annotation request to be made on images. The authors evaluate an annotation's influence by looking at the amount of entropy reduced on the labels of the entire data. Our formulation is distinct in

that it handles active learning of actions in video, and it includes a novel entropy metric computed in a space-time vote space.

Aside from active learning, researchers have also pursued interactive techniques to minimize manual effort in video annotation. This includes semi-automatic tools that make crowd-sourcing video annotation easier [35, 44], interactive segmentation methods for object labeling [37, 19], as well as using external meta-data like movie scripts to find potentially relevant content for a specified set of actions [14, 6]. While all share our goal of reducing the load on human labelers, our focus is to select which videos should even be labeled, not to make the labeling procedure itself easier.

Chapter 3

Approach

Given a small pool of labeled videos, our method initializes a voting-based action detector for the class of interest (e.g., an "answer phone" detector). Then we survey all remaining unlabeled videos, for which both the action labels and intervals are unknown, and identify the video whose annotation (if requested) is likely to most improve the current detector. To measure the value of each candidate video, we predict how it would reduce the uncertainty among all unlabeled videos. We gauge uncertainty by the entropy in the vote space that results when the current detector's training data is temporarily augmented to include the candidate's predicted action intervals. If our method works well, it will produce an accurate detector with minimal total manual annotations. Next we explain the details of each step.

3.1 Video Annotations

The few labeled videos used to initialize the action detector are annotated with both the temporal extent of the action as well as its spatial bounding box within each frame. In contrast, videos in the unlabeled pool are not trimmed to any specific action. This means they could contain multiple instances of one action and/or multiple different action classes.

When our system requests a manual annotation for a clip, it will get back two

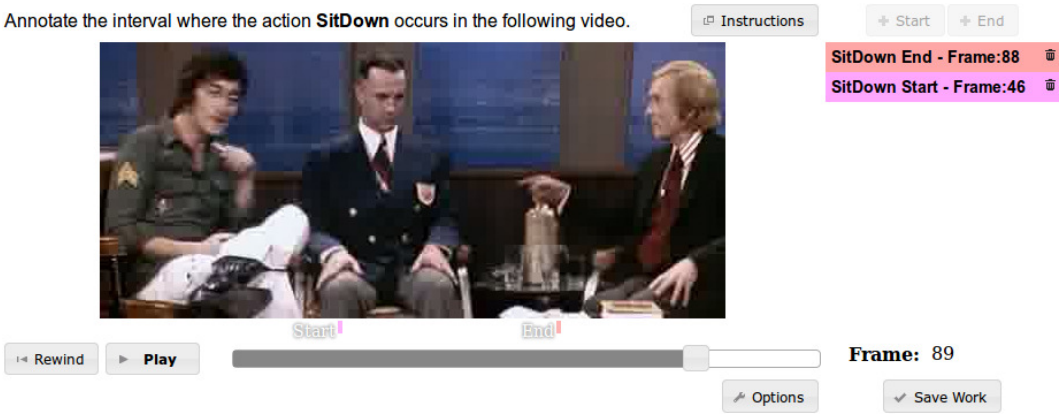


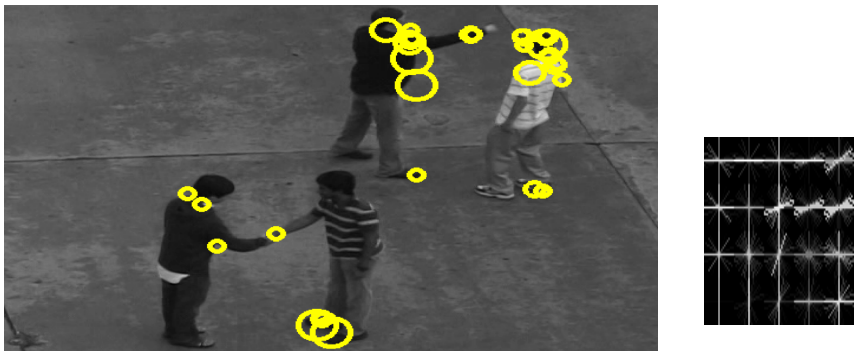
Figure 3.1: Our interface that annotators use to trim the actively requested videos.

pieces of information: 1) whether the action class of interest is present, and 2) if it is, the spatio-temporal subvolume where it is located. To aid in the latter, we enhanced the VATIC tool [35] to allow annotators to specify the temporal interval in which an action is located. The interface allows the user to play the video, jump around in time with a slider, and quickly place start and end points. See Figure 3.1. We will share our interface publicly to assist other researchers' labeling efforts.

3.2 Building the Action Detector

Our active learning approach requires an action detector as a subroutine. Taking into account our need for fast incremental updates with new training data, as well as our desire to detect actions with only partial evidence, we opt for a voting-based detector. We use local space-time features to vote on the localization parameters of the action, in a similar spirit to [38, 41]. In this section we explain how the detector is trained and updated; in the subsequent section we explain how it is applied to new data.

We first detect space-time interest points (STIPs) [13] (see Figure 3.2(a)) in the initial set of training videos $\mathcal{T}$. Then at each interest point, we extract a histogram-of-optical-flow (HoF) and histogram-of-oriented-gradients (HoG) feature [14]. HoG features



(a) Space-Time Interest Points (b) HoG features

Figure 3.2: Space-Time Interest Points (STIPs) detected on a frame and visualization of HoG features extracted in a STIP's neighborhood.

capture the local shape and appearance in space-time volumes in the neighborhood of detected STIPs and HoF features capture the optic flow (see Figure 3.2(b)). We cluster the HoG and HoF descriptors separately using k-means to build two visual vocabularies. Now every local feature is associated with two visual word indices w_{hog} and w_{hof}. Given these features, training the detector entails two main steps: 1) populating the Hough tables, and 2) prioritizing the votes of more discriminative words. See Figure 3.3 for an illustration of the training phase.

For the first step, we build one Hough table for HoG and one for HoF, denoted $\mathcal{H}_{hog}$ and $\mathcal{H}_{hof}$. The two tables are indexed by their respective visual vocabularies. For each local feature detected in the bounding volume of a positive training instance, we identify its associated visual words (the nearest k-means cluster centers w_{hog} and w_{hof}). Then, for those words, we add an entry in the Hough tables consisting of the 3D displacement vector between the local feature's (x, y, t) coordinates and the action's center.

For the second step, we rank the words in both tables according to their discriminative power for the action of interest. Following [38], we use precision and recall. Let

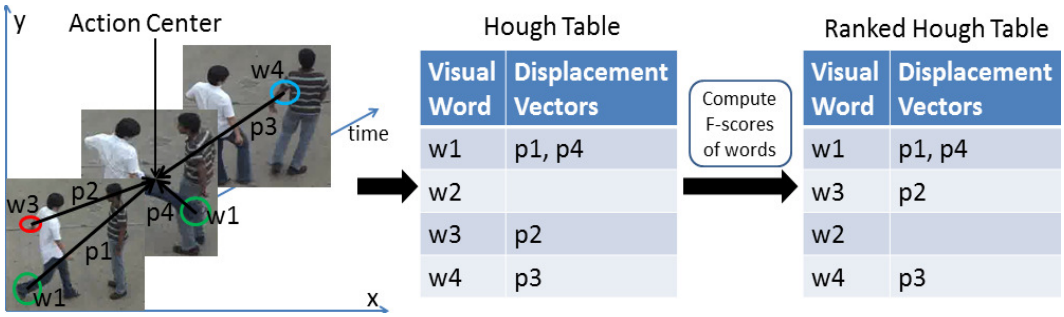


Figure 3.3: Overview of training of the detector.

$\mathcal{T}_p \subseteq \mathcal{T}$ consist of the positive labeled training intervals. For each HoG or HoF word w,

$$\text{PRECISION}(w) = \frac{|\mathcal{T}_p^w|}{|\mathcal{T}^w|}; \quad \text{RECALL}(w) = \frac{|\mathcal{T}_p^w|}{|\mathcal{T}_p|},$$

where $|\mathcal{T}_p^w|$ is the number of positive intervals containing word w, $|\mathcal{T}_p|$ is the number of positive intervals, and $|\mathcal{T}^w|$ is the total number of training examples with word w (whether positive or negative). We score each word by its F-measure, i.e., the harmonic mean of its precision and recall. We sort the HoG and HoF words separately by their F-measures, and maintain the sorted lists $\mathcal{W}_{hog}$ and $\mathcal{W}_{hof}$ as part of the detector. At test time, only the most discriminative words will cast votes.

Our full action detector D trained on data $\mathcal{T}$ consists of the Hough tables and sorted discriminative words: $D(\mathcal{T}) = (\mathcal{H}_{hog}, \mathcal{H}_{hof}, \mathcal{W}_{hog}, \mathcal{W}_{hof})$. Adding a newly labeled positive instance entails both expanding the Hough table and revising the F-scores; adding a negative entails revising the F-scores only. Thus, as active learning proceeds, certain words may gain or lose entry into the list of those that get to cast votes during detection.

3.3 Applying the Detector to a Novel Video

Given a novel test video, we detect any space-time subvolumes likely to contain the action. First, we extract STIPs and HoG/HoFs, and map them to visual words. Then, for any words

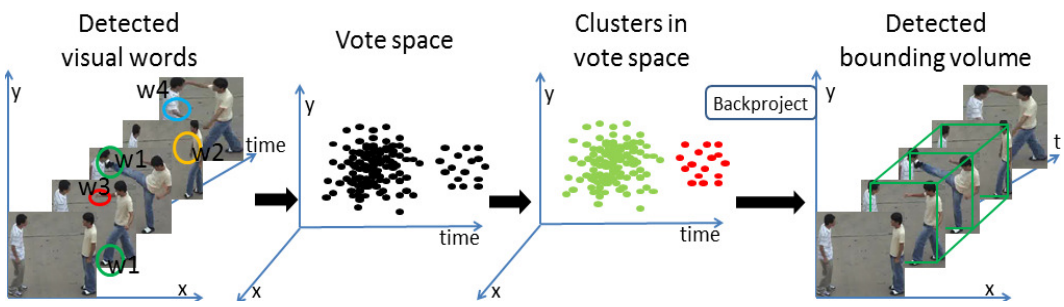


Figure 3.4: Overview of application of the detector on an unseen video.

appearing in the top $N = 100$ discriminative words in $\mathcal{W}_{hog}$ and $\mathcal{W}_{hof}$, we look up the corresponding entries in $\mathcal{H}_{hog}$ and $\mathcal{H}_{hof}$, and use those 3D displacement vectors to vote on the probable action center (x, y, t). We apply Mean-Shift clustering on the 3D vote space to discover the primary modes, discarding any clusters with fewer votes than the average cluster.

To obtain the bounding volume of the detected action, we backproject from these surviving vote clusters to features in the video. That is, we collect the 3D positions of all local features whose votes contributed to a given cluster. Then the coordinates of the detected action's bounding volume are determined by the minimum and maximum value of those backprojected positions in each dimension. See Figure 3.4 for an illustration of how the detector is applied on a novel video.

This voting procedure assumes that an action at test time will occur at roughly the same time-scale as *some* exemplar from training. It maintains some flexibility to time-scale variations, since any variation present in the training set is captured, and the clustering of votes into broad "bins" permits leeway in the precise alignment.

Finally, we sort the resulting detection hypotheses by the number of votes in the clusters that led to them. In the ideal case, a true detection would have nearly all votes cast near its action center. Hence, we treat vote count as a measure of detection confidence.

3.4 Active Selection of Untrimmed Videos

At this stage, we have an action detector $D(\mathcal{T})$, and an unlabeled untrimmed pool of video clips $\mathcal{U}$. The active learning loop iterates between examining the unlabeled data, requesting the most useful annotation, and updating the detector.

The key technical issue is how to identify the untrimmed video that, if annotated, would most benefit the detector. In active learning, the statistically optimal criterion is to select the unlabeled data point that leads to the greatest expected reduction in error on all unlabeled data once used to augment the training set, as doing so optimizes the metric that the classifier will ultimately be evaluated on [4, 25]. Motivated by this general idea, we define a criterion tailored to our problem setting that seeks the unlabeled video that, if used to augment the action detector, would most reduce the vote-space uncertainty across all unlabeled data. Specifically, the best video v to annotate is:

$$v^* = \underset{v \in \mathcal{U}}{\operatorname{argmax}} \quad \max_{l \in \mathcal{L}} S(\mathcal{T} \cup v^l), \tag{3.1}$$

where $\mathcal{L} = \{+1, -1\}$ is the set of possible labels, and v^l denotes that the video v has been given label l. $S(\cdot)$ is a scoring function that takes a training set as input, and estimates the confidence of a detector trained on that data and applied to all unlabeled data (to be defined next). Thus, the objective function considers how the detector might change after updating it with any possible next annotation.

With untrimmed videos, scoring confidence is non-trivial. In particular, if we were to consider v^l as a potential positive (denoted v^+) by simply updating the detector with vote vectors extracted from *all* features within it, we are likely to see no reduction in uncertainty—even if that video contains a great example of the action. The reason is that the features *outside* the true positive action interval would introduce substantial noise into the Hough tables.

We overcome this problem by first estimating any occurrences of the action within

v using the current detector and the procedure given in Sec. 3.3, and then predicting their individual impact. Suppose there are K detected occurrences in v. We use the bounding volume for each one in turn to temporarily augment the training set $\mathcal{T}$. This yields a series of K temporarily modified detectors $D(\mathcal{T} \cup \hat{v}_k^+)$, for $k = 1, \ldots, K$, where $\hat{v}_k$ denotes the k-th detection subvolume. We estimate the confidence value for the entire unlabeled dataset $\mathcal{U}$ according to each of these detectors in turn, and record the maximum value observed:

$$S(\mathcal{T} \cup v^+) = \max_{k=1,\ldots,K} \textsc{Value}\big(D(\mathcal{T} \cup \hat{v}_k^+)\big). \tag{3.2}$$

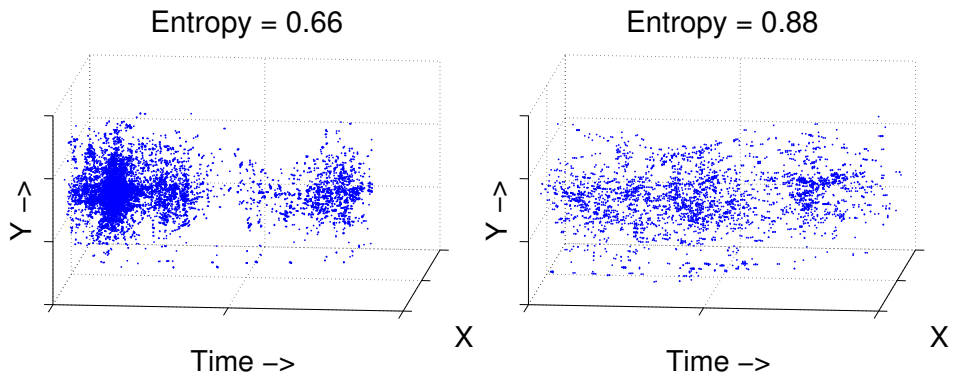
Since the unlabeled video may very well contain no positive intervals, we must also consider it as a negative instance. This yields a single modified detector in which v is introduced as a negative instance (denoted v^-), with the following confidence score:

$$S(\mathcal{T} \cup v^-) = \textsc{Value}\big(D(\mathcal{T} \cup v^-)\big). \tag{3.3}$$

Recall that positive intervals modify the detector in two ways: 1) by updating the Hough tables with votes for the new action centers, and 2) by updating the top N most discriminative words in $\mathcal{W}_{hog}$ and $\mathcal{W}_{hof}$. Negative videos modify the detector only in terms of the latter.

Note that this stage of selection is well-served by voting: whereas a naive approach would evaluate all possible spatio-temporal intervals, with voting we can efficiently hypothesize a small subset of candidates per video. Furthermore, the large number of temporary detector updates are efficiently handled by our method, since incrementally adding and removing displacement vectors from Hough tables are lightweight operations.

The final important piece is how to compute the VALUE function in Eqns. 3.2 and 3.3, which ought to reflect how well the detector can localize actions in every unlabeled video. To this end, we propose a entropy-based metric computed in the vote space. We use each candidate detector (modified as described above) to cast votes in each unla-



(a) High confidence (b) Low confidence

Figure 3.5: Example vote spaces for unlabeled videos. In 3.5(a), the entropy value is low because of good clusters around centers of the action. In 3.5(b), it is high, as there is no well-defined cluster in the vote space. Accordingly, we would predict high confidence for the video in 3.5(a) and low confidence for 3.5(b).

beled video $v \in \mathcal{U}$, using the procedure in Sec. 3.3. Each video v yields a set of continuous 3D votes V_v, where each vote is a (x, y, t) coordinate. Intuitively, a vote space with good cluster(s) indicates there is consensus on the location(s) of the action center, whereas spread out votes suggest confusion among the features about the action center placement.

To capture this, we quantize the vote space into uniformly sized 3D bins[1] and compute entropy. See Figure 3.5. Let c_b denote the count of votes in bin b. We define the probability that an action center occurs in bin b as $p(b) = \frac{c_b}{\sum_b c_b}$. Then the normalized entropy over the entire vote space V_v when using detector $D(\mathcal{T})$ is:

$$H\big(V_v, D(\mathcal{T})\big) = \frac{-\sum_b p(b)\log(p(b))}{\log B}, \tag{3.4}$$

where B denotes the total number of 3D bins.

Using this entropy-based uncertainty metric, we define the confidence of a detector

[1]In our implementation, we fix the size of the bins to be fairly coarse, at about one third of the spatial dimensions and about one second in the temporal dimension. We err on the side of large bins in order to catch candidates that might be useful.

in localizing actions on the entire unlabeled set:

$$\textsc{Value}\big(D(\mathcal{T})\big) = \frac{1}{|\mathcal{U}|}\sum_{v\in\mathcal{U}}\big(1 - H\big(V_v, D(\mathcal{T})\big)\big). \tag{3.5}$$

We stress that our method does *not* choose the individual video that has the lowest entropy. Rather, it selects the video which is *most expected to reduce entropy among all unlabeled videos*. The distinction is important. The former would be susceptible to selecting uninformative false positive videos, where even a small number of words agreeing strongly on an action center could yield a low entropy score. In contrast, our method is unlikely to choose such cases, as they are unlikely to reduce entropy across all the videos. (We will see this expected advantage play out in the results, in our comparison to an Active Entropy baseline.) Due to the nature of voting, spurious votes from bad examples are unlikely to agree. Essentially, our method helps discover true positive intervals, since once added to the detector they most help "explain" the remainder of the unlabeled data.

Our objective uses a $\max$ in Eqns. 3.1 and 3.2, as opposed to an expectation over all possible labels, making ours an "optimistic" [8] estimate of risk. In early experiments, we found this to be better in practice, likely because it avoids clouding the scoring function with noisy posteriors. Intuitively, our method predicts the "best case" impact of any unlabeled video. This means, for example, that a false positive interval in an unlabeled video will not overshadow the positive impact of a true positive interval in the same video.

To recap, the selection stage consists of optimizing Eqn. 3.1: we consider each unlabeled video in turn, and score its predicted influence on the uncertainty of all unlabeled videos. Each time a candidate video is used to introduce changes for the modified detectors (Eqn. 3.2 and 3.3), those changes are rolled back after the scoring function is computed. Finally, we request annotations for the most promising video, and use them to update the detector permanently. Then the process repeats. Figure 3.6 depicts the data flow.

On UT-Interaction dataset with an unlabeled pool of 34 videos, our algorithm takes roughly 40 minutes for selecting one candidate video for annotation. The run-time of our

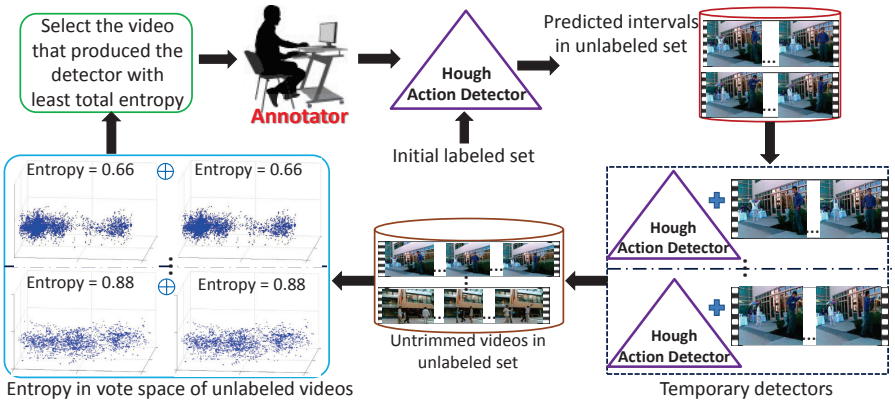


Figure 3.6: Overview of our active learning algorithm. We initialize our voting-based action detector with a small set of labeled videos (top center). Then we use the detector to predict intervals of the class of interest in the unlabeled videos (top right). We update the current detector temporarily with each interval predicted (bottom right), and use it to compute the entropy in the vote space of every unlabeled video. The total entropy in the unlabeled videos is then used to gauge the uncertainty reduced by the temporary detector (left bottom). For the next round of annotation, we select the video that produces the highest reduction in uncertainty, and update the current detector with the obtained annotation (top center). The cycle repeats with the new detector.

algorithm depends on the length of unlabeled videos, the number of features extracted from them and the clustering algorithm used for finding clusters in vote space. More significantly, it depends on the size of the unlabeled set since we evaluate the value of a temporary detector by examining the vote space of *all* unlabeled videos. Splitting up the untrimmed videos into smaller segments (using shot detection or at uniform intervals) such that they have at most 200-300 frames could be done to make it faster. Shorter videos have fewer features that vote for an action center, and less number of votes helps find clusters quickly.

Chapter 4

Experiments

We validate our approach on three datasets, and compare to several alternative selection strategies.

4.1 Datasets and Implementation Details

We use three public datasets: Hollywood [14], UT-Interaction [26], and MSR Actions 1 [43] (See Figure 4.1 for examples). These datasets have the spatio-temporal annotations needed for our experiments. Hollywood contains 8 action classes (AnswerPhone, GetOutCar, HandShake, HugPerson, Kiss, SitDown, SitUp, StandUp) and 430 total videos. We use the spatial annotations provided by [22] and improve the ground truth temporal trimming with our interface (Sec. 3.1). UT-Interaction contains 6 actions (HandShake, Hug, Kick, Point, Punch, Push) and 20 total videos. MSR Actions contains 3 actions (Clapping, Waving, Boxing) and 16 total videos. Since the latter two datasets have relatively few sequences, we split them into smaller segments with shot detection [1]. If an action is split between two shots, the shot with the major portion is assigned that action's label. This step yields 80 total videos for UT and 45 for MSR.

We set the vocabulary size $k = 1500$ for Hollywood and $k = 1000$ for UT and MSR. We set the Mean-Shift bandwidth as $1.2 \times \sqrt{F}$, where F denotes the number of

(a) Hollywood

(b) UT-Interaction

(c) MSR Action 1

Figure 4.1: Example frames from the three datasets used for our experiments.

frames in a video. We fix the bin size for entropy computation to $(40, 40, 30)$ pixels in the (x, y, t) dimensions. While Mean-Shift can return a variable number of modes, for efficiency during active selection we limit the number of candidates per video to $K = 2$.

We initialize all models with the same L random positive and negative labeled examples, and average results over 15 random sets. We set L as a function of the total positives available per action, yielding $L = 4$ for UT, $L = 3$ for MSR, and $L = 8$ for Hollywood. The exception is GetOutCar and SitUp in Hollywood, for which the dataset contains only 13 and 11 positive exemplars; for these classes we have only enough data to initialize with $L = 4$. For Hollywood we use the standard test split of 211 videos; for UT-Interaction and MSR Actions we take 38 and 18 test videos, respectively. The test set is constant for all iterations and runs.

4.2 Evaluation Metric

We quantify performance with learning curves: after each iteration, we score action localization accuracy on an unseen test set. Steeper and higher learning curves are better. We consolidate overlapping detections with the greedy algorithm in [38]. We score the top three detections per video, which is the maximum to appear in a test video. As is standard, we use overlap accuracy $\frac{A \cap B}{A \cup B}$ between a detection A and ground truth interval B. To be a true positive, a detection must overlap by at least $1/8$, following [43]. Below we report both raw overlap scores as well as the percentage of accuracy achieved when compared to a detector trained on the entire training set. The latter makes it easy to interpret how far a detector is from the maximum accuracy possible were all data labeled.

We run for 10-20 rounds, depending on the dataset size, at which point our method has typically discovered all positive instances.

4.3 Methods Compared

We test two variants of our approach: one that uses the true action intervals when evaluating a video for active selection (**Active GT-Ints**), and one that uses the (noisier) Hough-predicted intervals (**Active Pred-Ints**). Active-GT-Ints essentially shows the full potential of our method. It is valuable because it isolates the impact of the proposed active learning strategy from that of the specific detector we use.

Since no prior work does active learning for action classification (let alone detection), we designate four informative baselines that reveal the impact of our method design:

- **Passive** selects randomly from the unlabeled set.
- **Active Classifier** uses margin-based uncertainty sampling, a common active selection strategy [31]. We build an SVM action classifier using a χ^2 kernel and bag-of-words histograms computed for every bounding volume in the initial labeled videos. The classifier predicts the probability each unlabeled video is positive, and we request a label for the one whose confidence is nearest 0.5.
- **Active Entropy** is a detector-based method. This method selects the most uncertain video for labeling, where detector uncertainty is computed with the same vote space entropy in Eqn. 3.4.
- **Active MS-Ints** follows the approach of **Active Pred-Ints**, but uses high-density STIP volumes in a video, obtained by clustering the 3D locations of STIPs using Mean-Shift. The top 2 intervals (1 in case of Hollywood) with high STIP density are used in active selection.

All methods use the same features and Hough detector; thus, any differences in accuracy are attributable to the quality of their annotation requests.

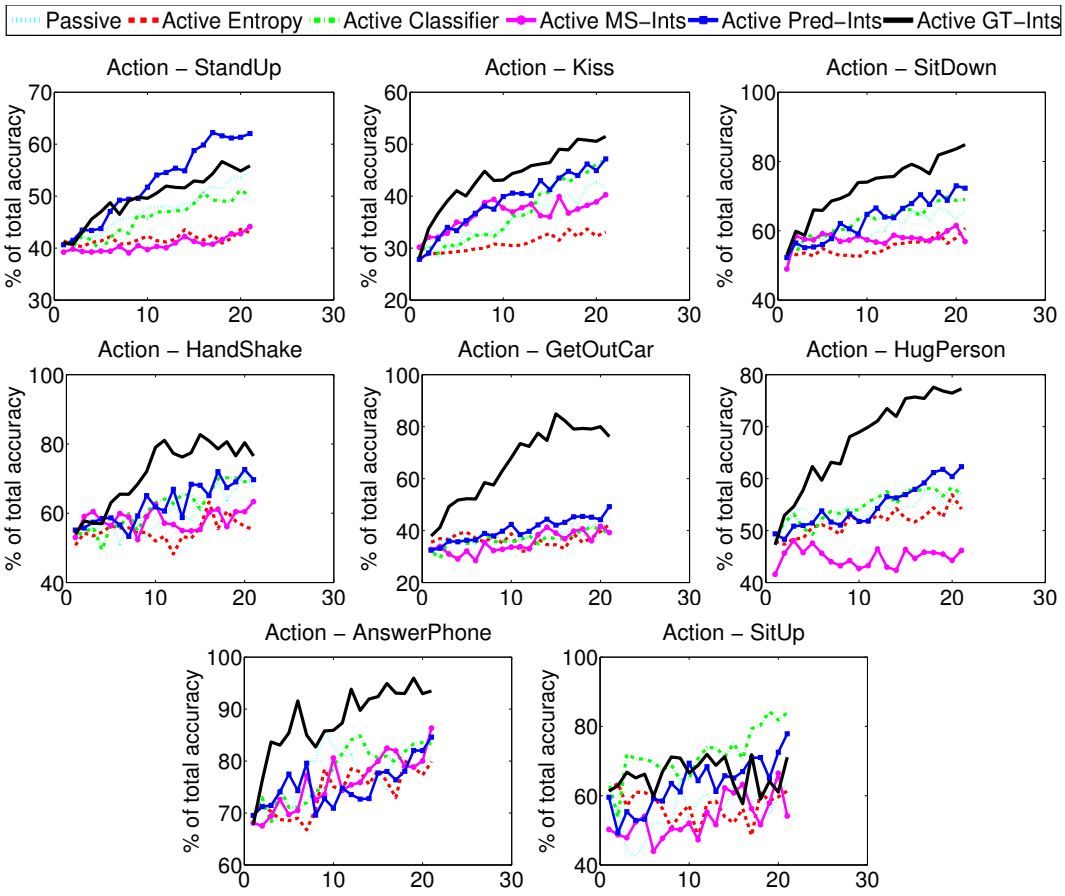


Figure 4.2: Results on the Hollywood dataset.

4.4 Results

Figure 4.2 shows results for the Hollywood actions.[1] In 7 of the 8 action classes, Active-GT-Ints clearly outperforms all the baseline methods. On SitUp, however, it is a toss-up, likely because parts of this action are very similar to the beginning and ending sequences of StandUp and SitDown, respectively, and hence the detector votes for action centers of other classes. When we use the predicted intervals, our accuracy decreases as expected, but is still strongest in more than half the cases. Looking more closely, we see that our

[1] In all results, the method's starting points can vary slightly due to the randomized Mean-Shift clustering step of the detector.

advantage is best for those classes where more videos are available in the dataset; this allows us to initialize the detectors with more positive samples ($L = 8$), and the stronger initial detectors make uncertainty reduction estimates more reliable. Notably, however, even with those same stronger detectors, the simpler active baselines underperform ours.

Figure 4.3 shows the results on the UT-Interaction dataset. Results are fairly consistent. Active GT-Ints is always strongest, showing the full potential of our active selection strategy. On the class Hug we see the biggest advantage for both variants of our method, likely because, unlike classes Kick and Push, there are just a few distinct variations between different executions of Hug. Our active detector quickly explores the distinct useful types for annotation. It is also the reason for the highest recognition accuracy for Hug when compared to other classes. However, our predicted intervals are weak on the class Point. In this action, the actor stands still and points at something, which leads to fewer STIPs in the action bounding volume, and thus an insufficient set of discriminative words for voting.

We find the Hough detector occasionally generates large clusters around action centers of a different action class. This can mislead the entropy computation and result in low entropy values for negative videos. While negative videos can still be informative to refine the words' F-measures, typically positive instances yield faster payoff by expanding the Hough tables. This suggests that there is scope for improvement in the discriminative aspect of our detector.

Figure 4.4 shows the MSR Actions results. We see the advantage of our approach early in the learning curves for Clapping and Boxing, getting the biggest boost with the least annotator input. The periodic nature of the MSR Actions poses a challenge for all methods. Periodicity is problematic for voting because the features specific to the action re-occur at multiple places in the bounding volume. At test time, this causes confusion in the vote space, since the same word casts many diverging votes.

Throughout our experiments, none of the baselines perform consistently well for all actions. The Active Entropy baseline is often the weakest. It selects mostly negative

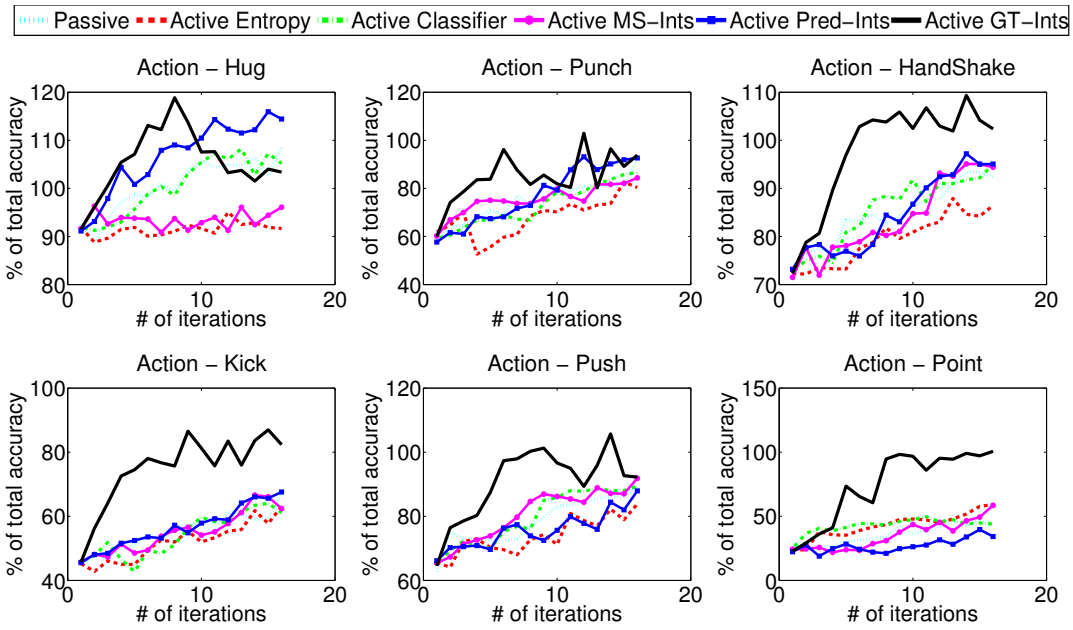


Figure 4.3: Results on the UT-Interaction dataset.

examples for labeling, and does not work as well as uncertainty-based techniques do in traditional settings. This supports our claim that simply estimating *individual* video uncertainty is insufficient for our setting; our use of total entropy reduction over all unlabeled data is more reliable. Interestingly, the Active Classifier baseline often underperforms the Passive approach, again due to its failure to find positive samples for labeling. This underscores the importance of reasoning about the untrimmed nature of video when performing active learning.

The Active MS-Ints baseline uses intervals with a high volume of STIPs as interval predictions. The presence of STIPs is usually an indication of motion in a video. So the volumes enclosing these STIPs provide good estimates of action intervals. Using these intervals in our active selection algorithm provides us with a reasonable baseline approach to gauge the impact of the interval predictions (made by our detector) used in Active Pred-Ints. The Active MS-Ints baseline performs poorly in most of the cases when compared to Active Pred-Ints (especially on Hollywood). Since the only difference between these two methods

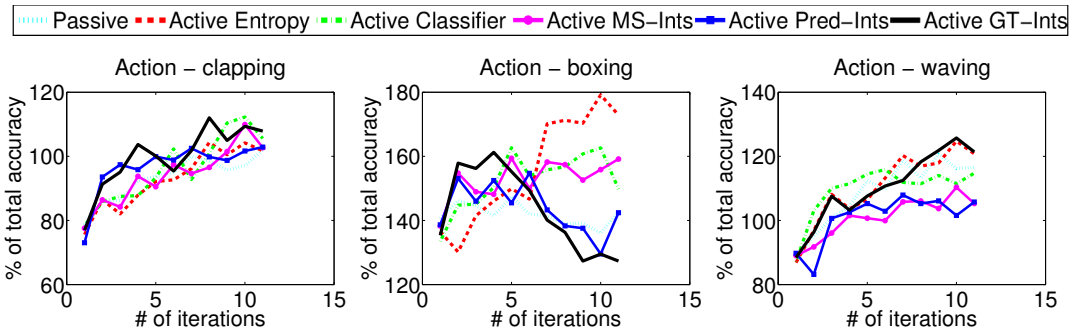


Figure 4.4: Results on the MSR Actions dataset.

is the interval predictions used in active selection, we can conclude that the stronger learning curves for Active Pred-Ints are a direct result of the better interval predictions made by our voting-based detector.

The difference between Active GT-Ints and Active Pred-Ints reveals that improved interval prediction will have the most impact on our results, and so future work should focus on this aspect. Nonetheless, it is clear that the voting-based intervals we predict are better than simpler alternatives.

Figure 4.5 shows Active Pred-Int's mean overlap accuracy per dataset at three different points: 1) at the onset, using the small initial labeled set, 2) after 10-20 rounds of active learning, and 3) after adding *all* the videos with their annotations. Typically the active approach produces a detector nearly as accurate as the one trained with all data, yet it costs substantially less annotator time, e.g., using as little as 12% of the total annotations for the largest dataset, Hollywood.

In some cases, the actively chosen annotations yield a detector that actually generalizes *better* at test time than the one trained with all possible examples. Hence our overlap plots in Figures 4.3, 4.4 can have overlap accuracy greater than 100%. This suggests that the role of intelligent selection may go beyond simply saving annotator time; rather, it has potential to eliminate data that even if labeled is not useful. Again, this is an upshot of formulating the selection criterion in terms of total uncertainty reduction on all unlabeled data.

Train set	SitDown	HandShake	StandUp	AnswerPhone	GetOutCar	HugPerson	Kiss	SitUp	Mean
Initial L ex only	0.0918	0.0556	0.0988	0.0778	0.0446	0.1295	0.0411	0.0560	0.0744
After 20 rounds active	0.1271	0.0702	0.1508	0.0947	0.0677	0.1635	0.0696	0.0733	0.1021
Full train set (219 ex)	**0.1758**	**0.1008**	**0.2431**	**0.1119**	**0.1375**	**0.2624**	**0.1475**	**0.0941**	**0.1591**

(a) Hollywood

Train set	HandShake	Hug	Kick	Point	Punch	Push	Mean
Initial L ex only	0.1981	0.3029	0.1466	0.0107	0.1094	0.2022	0.1616
After 15 rounds active	0.2574	**0.3804**	0.2175	0.0164	0.1758	0.2689	0.2194
Full train set (42 ex)	**0.2708**	0.3324	**0.3218**	**0.0478**	**0.1897**	**0.3058**	**0.2447**

(b) UT-Interaction

Train set	Clapping	Waving	Boxing	Mean
Initial L ex only	0.2288	0.2318	**0.1135**	0.1914
After 10 rounds active	**0.3379**	**0.3134**	0.1043	**0.2519**
Full train set (27 ex)	0.3132	0.2582	0.0819	0.2178

(c) MSR Actions

Figure 4.5: For each dataset, we compare the detector's initial accuracy (first row per table), its accuracy after active learning with our method (middle row per table), and its accuracy if all available training instances are used (last row per table). By focusing annotation effort on only the most useful 10-20 videos, our method yields a detector nearly as accurate as the one trained with all data.

Rather than request labels on individual videos that look uncertain, our method searches for examples that help explain the plausible detections in all remaining unlabeled examples.

To give a concrete sense of the improvement our method offers, we analyze the significance of points in overlap accuracy. Suppose a video has one ground truth interval of time-length 50 and window size 300×300. Final spatio-temporal overlap gains of 2-3 points indicate a detection that correctly identifies 10-18% more of the frames for the action, or 5-10% more of correct pixels in the spatial dimension, or some combination thereof. Thus, the differences between the baselines (shown in the learning curves) and our raw gains relative to the initial models (shown in the tables) are noticeable in practical terms. Figure 4.6 shows some example detections alongside ground truth.

Figure 4.6: Detections for Clapping (top row), Waving (middle), and Boxing (bottom) from MSR Actions. The bounding boxes of our detections (red) align closely with those of the ground truth (green), after only 15 rounds of annotation.

Chapter 5

Conclusion

Untrimmed video is what exists "in the wild" before any annotators touch it, yet it is ill-suited for traditional active selection metrics. We introduced an approach to discover informative action instances among such videos. Our main idea is to use a voting-based action detector to localize likely intervals of interest in an unlabeled clip, and use them to estimate the value of annotating that unlabeled clip. The value of annotating a video is measured by the total reduction in uncertainty of all unlabeled videos that it would produce if it were labeled. We also introduced a novel technique to compute the uncertainty of an action detector by finding entropy in the 3D vote space of the video. We show that a selection method that accounts for the untrimmed nature of videos can help us make better decisions in selecting clips for annotation, and hence produce detectors that are better when compared to the ones built by selecting videos using simple active selection techniques. Our results also demonstrate that active learning with a few training sample could sometimes produce detectors that can generalize better at test time than the ones built using the entire training set. Our results show the potential to reduce human annotation effort and produce more reliable detectors with well-chosen examples.

5.1 Future Work

Since the accuracy of the predicted intervals used in the active selection phase is paramount to the success of our selection algorithm, we are interested in exploring alternate detection strategies that can produce more accurate interval predictions. Also, multi-class active selection is another avenue for future work that can reduce a lot of computation time by evaluating the informativeness of an unlabeled video for multiple classes in a single shot.

The videos we use in this work are from challenging state-of-the-art action recognition/detection datasets. While for some of the datasets we automatically break the original videos down into shorter ones, those shorter videos also contain multiple actions, and we do parse the sub-parts of each video for evaluation. In general, an interesting future direction is to explore alternative ways to pre-process long videos to ”undersegment” them in an action class-independent manner, prior to running the detector. This is due to the fewer number of actions one has to evaluate in a video, shorter processing time for each video and better usability with the annotation tools. As mentioned previously, making the detector better and more discriminative also helps in this regard. These are some possible interesting and challenging directions for future work in this domain.

Bibliography

[1] mmlab.disi.unitn.it/wiki/index.php/Shot_Boundary_Detection.

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

[3] C. Chen and K. Grauman. Efficient Activity Detection with Max-subgraph Search. In *CVPR*, 2012.

[4] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *JAIR*, 4:129–145, 1996.

[5] B. Collins, J. Deng, K. Li, and L. Fei-Fei. Towards Scalable Dataset Construction: An Active Learning Approach. In *ECCV*, 2008.

[6] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic Annotation of Human Actions in Video. In *CVPR*, 2009.

[7] A. Fathi, M. Balcan, X. Ren, and J. Rehg. Combining Self Training and Active Learning for Video Segmentation. In *BMVC*, 2011.

[8] Y. Guo and R. Greiner. Optimistic active learning using mutual information. In *IJCAI*, 2007.

[9] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, 2007.

[10] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *SGA Wkshp*, 2010.

[11] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively selecting annotations among objects and attributes. In *ICCV*, 2011.

[12] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A Large Video Database For Human Motion Recognition. In *ICCV*, 2011.

[13] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[14] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions From Movies. In *CVPR*, 2008.

[15] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization And Segmentation With An Implicit Shape Model. In *ECCV*, 2004.

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

[17] A. Oikonomopoulos, I. Patras, and M. Pantic. An Implicit Spatiotemporal Shape Model For Human Activity Localization And Recognition. In *CVPR*, 2008.

[18] V. Parameswaran and R. Chellappa. Human Action-Recognition using Mutual Invariants. *CVIU*, 2005.

[19] B. Price, B. Morse, and S. Cohen. Livecut: Learning-based Interactive Video Segmentation by Evaluation of Multiple Propagated Cues. In *ICCV*, 2009.

[20] G. Qi, X. Hua, Y. Rui, J. Tang, and H. Zhang. Two-dimensional active learning for image classification. In *CVPR*, 2008.

[21] D. Ramanan and D. Forsyth. Automatic Annotation of Everyday Movements. In *NIPS*, 2003.

[22] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012.

[23] K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications Journal*, 2012.

[24] M. D. Rodriguez, J. Ahmed, and M. Shah. Action Mach: A Spatio-Temporal Maximum Average Correlation Height Filter for Action Recognition. In *CVPR*, 2008.

[25] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, 2001.

[26] M. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, 2010.

[27] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *ECCV*, 2010.

[28] C. Schuldt, I. Laptev, and B. Caputo. Recognizing Human Actions: A Local Svm Approach. In *ICPR*, 2004.

[29] B. Settles. Active learning literature survey, 2009. Computer Sciences Technical Report 1648, Univ. of Wisconsin Madison.

[30] B. Siddiquie and A. Gupta. Beyond Active Noun Tagging: Modeling Contextual Interactions for Multi-Class Active Learning. In *CVPR*, 2010.

[31] S. Tong and E. Chang. Support Vector Machine Active Learning for Image Retrieval. In *ACM Multimedia*, 2001.

[32] S. Vijayanarasimhan and K. Grauman. Multi-level Active Prediction Of Useful Image Annotations For Recognition. In *NIPS*, 2009.

[33] S. Vijayanarasimhan and K. Grauman. Large-Scale Live Active Learning: Training Object Detectors With Crawled Data and Crowds. In *CVPR*, 2011.

[34] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *ECCV*, 2012.

[35] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently Scaling Up Crowdsourced Video Annotation. *IJCV*, 2012.

[36] C. Vondrick and D. Ramanan. Video Annotation and Tracking with Active Learning. In *NIPS*, 2011.

[37] J. Wang, P. Bhat, R. Colburn, M. Agrawala, and M. Cohen. Interactive Video Cutout. In *SIGGRAPH*, 2005.

[38] G. Willems, J. H. Becker, T. Tuytelaars, and L. V. Gool. Exemplar-based Action Recognition In Video. In *ECCV*, 2009.

[39] S. Wong and R. Cipolla. Extracting Spatiotemporal Interest Points Using Global Information. In *ICCV*, 2007.

[40] R. Yan, J. Yang, and A. Hauptmann. Automatically Labeling Data Using Multi-Class Active Learning. In *ICCV*, 2003.

[41] A. Yao, J. Gall, and L. V. Gool. A Hough Transform-based Voting Framework for Action Recognition. In *CVPR*, 2010.

[42] G. Yu, J. Yuan, and Z. Liu. Unsupervised Random Forest Indexing for Fast Action Search. In *CVPR*, 2011.

[43] J. Yuan, Z. Liu, and Y. Wu. Discriminative Subvolume Search for Efficient Action Detection. In *CVPR*, 2009.

[44] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme Video: Building a video database with human annotations. In *ICCV*, 2009.