Dynamic Pooling for Complex Event Recognition

Weixin Li[†] Qian Yu[§] Ajay Divakaran[§] Nuno Vasconcelos[†]

[†]University of California, San Diego
La Jolla, CA 92093, United States
{wel017, nvasconcelos}@ucsd.edu

[§]SRI International Sarnoff
Princeton, NJ 08540, United States
{qian.yu, divakaran.ajay}@sri.com

Abstract

The problem of adaptively selecting pooling regions for the classification of complex video events is considered. Complex events are defined as events composed of several characteristic behaviors, whose temporal configuration can change from sequence to sequence. A dynamic pooling operator is defined so as to enable a unified solution to the problems of event specific video segmentation, temporal structure modeling, and event detection. Video is decomposed into segments, and the segments most informative for detecting a given event are identified, so as to dynamically determine the pooling operator most suited for each sequence. This dynamic pooling is implemented by treating the locations of characteristic segments as hidden information, which is inferred, on a sequence-by-sequence basis, via a large-margin classification rule with latent variables. Although the feasible set of segment selections is combinatorial, it is shown that a globally optimal solution to the inference problem can be obtained efficiently, through the solution of a series of linear programs. Besides the coarse-level location of segments, a finer model of video structure is implemented by jointly pooling features of segment-tuples. Experimental evaluation demonstrates that the resulting event detector has state-of-the-art performance on challenging video datasets.

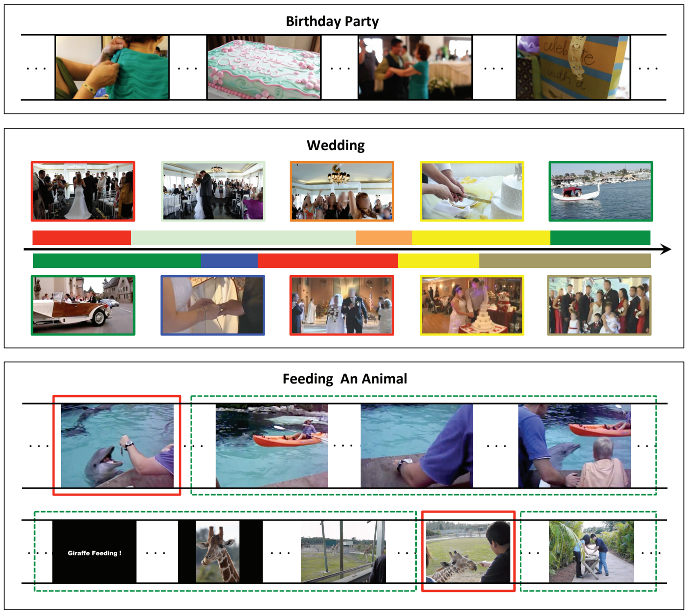

Figure 1: Challenges of event recognition in open source video (best viewed in color). An event class, *e.g.*, "birthday party", can involve a complex sequence of actions, such as "dressing", "cake cutting", "dancing" and "gift opening". Two instances of an event class, *e.g.*, "wedding", can differ substantially in the atomic actions that compose them and corresponding durations (indicated by color bars). For example, the upper "wedding" video includes the atomic actions "walking the bride" (red), "dancing" (light grey), "flower throwing" (orange), "cake cutting" (yellow) and "bride and groom traveling" (green). On the other hand, the lower "wedding" video includes the actions "ring exchange" and "group pictures" but no "dancing" or "flower throwing". Finally, a video depicting an event can contain contents unrelated to the event. In the "feeding an animal" examples, only a small portion (red box) of the video actually depicts the action of handing food to an animal. The location of this characteristic behavior can also vary significantly from video to video.

1. Introduction

The recognition of complex events in open source videos, *e.g.*, from YouTube, is a subject of increasing attention in computer vision [17, 5, 23, 16]. Unlike the recognition of primitive or atomic actions, such as "walking", "'running", from carefully assembled video, complex events depict human behaviors in unconstraint scenes, performing more sophisticated activities, which involve more complex interactions with the environment, *e.g.*, a "wedding ceremony", a "parade" or a "birthday party". In general, this kind of video is captured and edited by ama-

This work has been partially supported by NSF under Grant CCF-0830535 and NSF IIS-1208522, and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20066. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/NBC, or the U.S. Government.

teur videographers (*e.g.*, YouTube users), with little uniformity in terms of equipment, scene settings (view-point, backgrounds, *etc*), and mostly without professional post-processing, *e.g.*, video cutting, segmentation or alignment.

Due to all these, the detection of complex events presents two major challenges beyond those commonly addressed in the action recognition literature. The first is that the video is usually not precisely segmented to include only the behaviors of interest. For example, as shown in Figure 1, while the event "feeding an animal" is mostly about the behavior of handing the animal food, a typical YouTube video in this class depicts a caretaker approaching the animal, playing with it, checking its health, *etc*. The second challenge is that the behaviors of interest can have a complex temporal structure. In general, a complex event can have multiple such behaviors and these can appear with great variability of temporal configurations. For example, the "birthday party" and "wedding" events of Figure 1, have significant variation in the continuity, order, and duration of characteristic behaviors such as "walking the bride," "dancing," "flower throwing," or "cake cutting".

In the action recognition literature, the popular *bag of (visual) features* (BoF) representation has been shown to 1) produce robust detectors for various classes of activities [13, 27], and 2) serve as a sensible basis for more sophisticated representations [17, 23, 9, 14]. One operation critical for its success is the pooling of visual features into a holistic video representation. However, while fixed pooling strategies, such as average pooling or temporal pyramid matching, are suitable for carefully manicured video, they have two strong limitations for complex event recognition. First, by integrating information in a pre-defined manner, they cannot adapt to the temporal structure of the behaviors of interest. As illustrated with the "wedding" and "feeding an animal" examples of Figure 1, this structure is usually very rich and flexible for complex events. Second, by pooling features from video regions that do not depict characteristic behaviors, they produce noisy histograms, where the feature counts due to characteristic behavior can be easily overwhelmed by those due to uninformative content.

In this work, we address both limitations by proposing a pooling scheme adaptive to the temporal structure of the particular video to recognize. The video sequence is decomposed into segments, and the most informative segments for detection of a given event are identified, so as to *dynamically* determine the pooling operator *most suited for that particular video sequence.* This *dynamic pooling* is implemented by treating the locations of the characteristic segments as hidden information, which is inferred, on a *sequence-by-sequence basis*, via a large-margin classification rule with latent variables. While this entails a combinatorial optimization, we show that an *exact* solution can be obtained *efficiently,* by solving a series of linear programming. In this way, only the portions of the video informative about the event of interest are used for its representation.

The proposed pooling scheme can be seen either as 1) a discriminant form of segmentation and grouping, which eliminates histogram noise due to uninformative content, or 2) a discriminant approach to modeling video structure, which automatically identifies the locations of behaviors of interest. It is shown that this modeling can have different levels of granularity, by controlling the structure of the hypothesis space for the latent variable. Besides the coarse-level location of segments, finer modeling of structure can be achieved by jointly pooling histograms of segment-tuples. This is akin to recent attempts at modeling the short-term temporal layout of simple actions [9], but relies on adaptively rather than manually specified video segments. Experiments demonstrate that the detector significantly outperforms existing models of video structure.

2. Related Work

There has, so far, been limited work on pooling mechanisms for complex event detection. Laptev *et al.* extend spatial pyramid matching to the video domain and propose a BoF temporal pyramid (BoF-TP) matching for atomic action recognition in movie clips [13]. More recently, Cao *et al.* use unsupervised clustering of image features to guide feature pooling at the image level [5]. Since these pooling schemes cannot 1) select informative video segments, or 2) model the temporal structure of the underlying activities, they have limited applicability to complex event modeling. More broadly, the proposed method can be seen as a dynamic counterpart to recent advances in receptive field learning for image analysis [10]. While [10] assumes that the optimal spatial regions (receptive fields) for pooling descriptors of a given category are fixed, our work addresses content-driven pooling regions, *dynamically* or *adaptively* discovered on a sequence-by-sequence basis.

Several works have addressed the modeling of temporal structure of human activities. These can be grouped in two major classes. The first class aims to capture the most discriminative subsequence for simple action recognition. Nowozin *et al.* [18] use boosting to learn a classifier that searches for discriminative segments. In [21], Schindler and Gool show that simple actions can be recognized almost instantaneously, with a signature video segment less than 1 second long. Similarly, Satkin and Hebert [20] explore the impact of temporally cropping training videos on action recognition. While starting to address the problem that we now consider, these methods have various limitations, *e.g.*, 1) ignoring the temporal structure within subsequences, 2) limiting the hypothesis space of video cropping to continuous subsequences (which precludes temporally disconnected subsequences that are potentially more discriminant for complex event recognition), and 3) limited

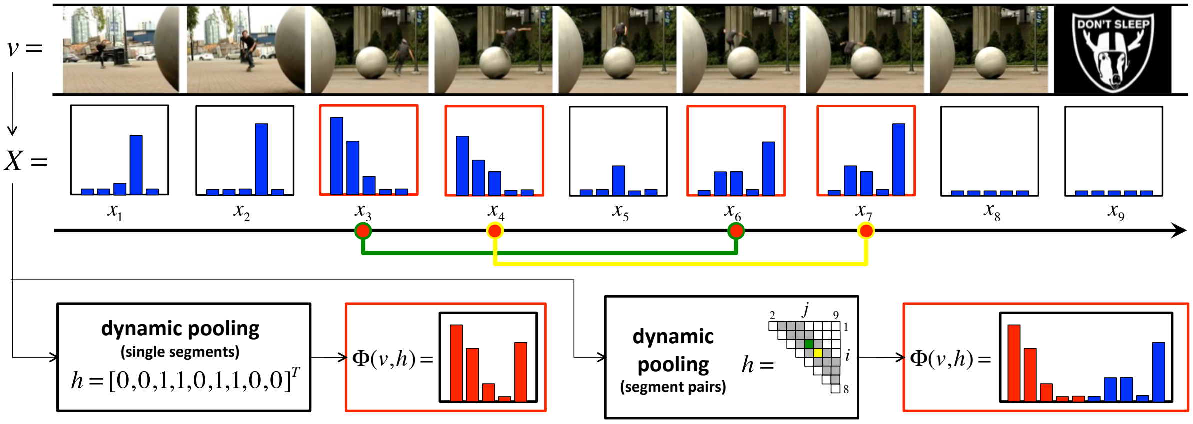

Figure 2: Dynamic pooling for recognizing "apply a board trick". The event is identified by signature actions of "jumping up with board" and "landing", which are mined out to represent the event either by segment or segment-pair pooling. Note that, in the segment-pair pooling, the feasible pairs are constraint by $L_1 = 2$ and $L_2 = 4$ in (12), as shown by the shaded elements in the triangular for pair (i, j).

ability to cope with the exponential nature of the hypothesis space (using heuristics to search for sub-optimal solutions). We address this problem by proposing an efficient procedure to dynamically determine the most discriminant segments for video classification.

The second class aims to factorize activities into sequences of atomic behaviors, and characterize their temporal dependencies [17, 8, 4, 23, 24, 14, 16]. While a number of representations have been proposed, *e.g.*, the spatio-temporal graphs of [4, 24], most methods are based on the BoF. Aiming to move beyond the BoF-TP of [13], Niebles *et al.* [17] and Gaidon *et al.* [8] raise the semantics of the representation, explicitly characterizing activities as sequences of atomic actions (*e.g.*, "long-jump" as a sequence of "run", "jump" and "land"). Li and Vasconcelos extend this idea by characterizing the dynamics of action attributes, using a binary dynamic system (BDS) to model trajectories of human activity in attribute space [14], and then to bag of words for attribute dynamics (BoWAD) [16]. Some drawbacks of these approaches include the need for manual 1) segmentation of activities into predefined atomic actions, or 2) annotation of training sets for learning attributes or atomic actions. Some automated methods have, however, been proposed for discovery of latent temporal structure. In particular, Tang *et al.* use a variant of the variable-duration hidden Markov model (VD-HMM) to learn both hidden action states and their duration [23]. Most methods in this group assume that 1) the entire video sequence is well described by the associated label, and 2) video sequences are precisely cropped and aligned with activities of interest. This is usually not the case for open source videos.

3. Event Detection via Dynamic Pooling

In this section we introduce a detector of complex events using dynamic pooling.

3.1. Complex Events

A complex event is defined as an event composed of several local behaviors. A video sequence $\boldsymbol{v}$ is first divided into a series of short-term temporal segments $\mathcal{S} = \{\boldsymbol{s}_i\}_{i=1}^{\tau}$, which are denoted *atomic segments*. This can be done with a sliding window, or algorithms for detecting shot transitions. Each of the segments $\boldsymbol{s}_i$ depicts a short-term behavior, characterized by a visual feature $\boldsymbol{x}_i \in \mathbb{R}^D$, *e.g.*, a histogram of visual word counts [13, 28]. A complex event is a segment subset $\bar{\mathcal{S}} \subseteq \mathcal{S}$, *i.e.*, an element of the power set of $\mathcal{S}$. Note that this does not have to be a continuous subsequence of $\boldsymbol{v}$ (as in [20]), but can be any combination of elements from $\mathcal{S}$ to allow for a more flexible representation.

3.2. Dynamic Pooling

Given the feature vectors $\boldsymbol{x}_i$ extracted from τ atomic segments $\boldsymbol{s}_i$ of sequence $\boldsymbol{v}$, a *holistic feature* is defined as

$$\boldsymbol{\Phi}(\boldsymbol{v}, \boldsymbol{h}) = \frac{X\boldsymbol{h}}{\boldsymbol{d}^T\boldsymbol{h}}, \tag{1}$$

where $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_\tau] \in \mathbb{R}^{D\times\tau}$ is a matrix whose i-th column is the feature vector extracted from $\boldsymbol{s}_i$, $\boldsymbol{d} \in \mathbb{R}^{\tau}_{++}$ a vector of positive segment confidence scores, and $\boldsymbol{h} \in \{0,1\}^\tau$ the indicator vector of the subset $\bar{\mathcal{S}}$, *i.e.*, $h_i = 1$ if $\boldsymbol{s}_i \in \bar{\mathcal{S}}$ and $h_i = 0$ otherwise. The feature $\boldsymbol{\Phi}(\boldsymbol{v}, \boldsymbol{h})$ can have different interpretations depending on the choice of features $\boldsymbol{x}_i$ and scores d_i. For example, when $\boldsymbol{x}_i$ and d_i are the unnormalized BoF histogram (*i.e.*, visual word counts) and number of visual features of i-th segment, respectively, $\boldsymbol{\Phi}(\boldsymbol{v}, \boldsymbol{h})$ is a BoF histogram over the subset $\bar{\mathcal{S}}$. This is illustrated by Fig. 2. Note that the L-1 normalization of (1) has been shown important for histogram-based large-margin classification [26]. By determining the composition of the subset $\bar{\mathcal{S}}$, $\boldsymbol{h}$ controls the temporal pooling of visual word counts. A fixed $\boldsymbol{h}$ implements a static pooling mechanism, *e.g.*, pyramid matching [13]. In this work, we introduce a *dynamic pooling* operator, by making $\boldsymbol{h}$ a latent variable, adapted to each sequence so as to maximize classification accuracy. This is implemented with recourse to a latent large-margin classifier.

3.3. Prediction Rule

A detector for event class c is implemented as $d(\boldsymbol{v}) = \text{sign}[f_{\boldsymbol{w}}(\boldsymbol{v})]$, where $f_{\boldsymbol{w}}(\boldsymbol{v})$ is a linear predictor that quantifies the confidence with which $\boldsymbol{v}$ belongs to c. This is implemented as

$$f_{\boldsymbol{w}}(\boldsymbol{v}) = \max_{\boldsymbol{h}\in\mathcal{H}} \left[\boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{v}, \boldsymbol{h}) + r(\boldsymbol{h})\right], \tag{2}$$

where $\boldsymbol{w} \in \mathbb{R}^D$ is a vector of predictor coefficients, $\boldsymbol{\Phi}(\boldsymbol{v}, \boldsymbol{h}) \in \mathbb{R}^D$ the feature vector of (1), $\boldsymbol{h}$ the vector of latent variables, $\mathcal{H}$ the hypothesis space $\{0,1\}^\tau$, and $r(\boldsymbol{h})$ a reward

$$r(\boldsymbol{h}) = r(||\boldsymbol{h}||_1) \tag{3}$$

with $r(\cdot)$ a non-decreasing function, which encourages configurations of $\boldsymbol{h}$ that use larger numbers of atomic segments to explain $\boldsymbol{v}$ as the complex event c. In this work, we adopt

$$r(\boldsymbol{h}) = a \log(||\boldsymbol{h}||_1) + b\,, \tag{4}$$

where $a \in \mathbb{R}_+$ and $b \in \mathbb{R}$ are parameters, but any other non-increasing function could be used in (3).

Note that (2) has two possible interpretations. Under the first, (4) is a bias term of the predictor, whose parameters are learnt during training. Under the second, (2) is a *maximum a posteriori* (MAP) prediction for a (log-linear) Bayesian model of 1) class conditional distribution proportional to $e^{\boldsymbol{w}^T\boldsymbol{\Phi}(\boldsymbol{v},\boldsymbol{h})}$ and 2) prior (on latent variable configurations) proportional to $e^{r(\boldsymbol{h})}$. In this case, a, b are fixed hyperparameters, encoding prior knowledge on event structure.

3.4. Inference

Given a sequence $\boldsymbol{v}$ and the parameters $\boldsymbol{w}, a, b$, the prediction of (2) requires the solution of

$$\text{(NLIP)} : f_{\boldsymbol{w}}(\boldsymbol{v}) = \max_{\boldsymbol{h}\in\mathcal{H}} \left[\frac{\boldsymbol{w}^T X \boldsymbol{h}}{\boldsymbol{d}^T \boldsymbol{h}} + r(||\boldsymbol{h}||_1)\right]. \tag{5}$$

Since the variable $\boldsymbol{h} \in \mathcal{H}$ is discrete, (5) is a *non-linear integer programming* (NLIP) problem and NP-hard under general settings. However, since $\boldsymbol{d} \in \mathbb{R}^\tau_{++}$, it can be solved efficiently, via the solution of a finite number of linear programming problems. This follows from two observations. The first is that (5) can be factorized as

$$f_{\boldsymbol{w}}(\boldsymbol{v}) = \max_{1\leqslant k\leqslant\tau, k\in Z} \Big[f^*(\boldsymbol{v}; \boldsymbol{w}, k) + r(k)\Big], \tag{6}$$

where $f^*(\boldsymbol{v}; \boldsymbol{w}, k)$ is the optimum of

$$\text{(ILFP)} : \quad \max_{\boldsymbol{h}\in\mathcal{H}} \frac{\boldsymbol{w}^T X \boldsymbol{h}}{\boldsymbol{d}^T \boldsymbol{h}}, \quad s.t. \quad \sum\nolimits_i h_i = k, \tag{7}$$

with $\boldsymbol{h}^*(k)$ as the optimal solution. This is an *integer linear-fractional programming* (ILFP) problem. The second observation is the following result.

Theorem 1 ([15]) *If $\boldsymbol{d} \succ 0$ (i.e., $\forall i$, d_i is strictly positive), the optimal value of (7) is identical to that of the relaxed problem*

$$(LFP) : \quad \max_{\boldsymbol{h}\in\mathcal{B}^\tau} \frac{\boldsymbol{w}^T X \boldsymbol{h}}{\boldsymbol{d}^T \boldsymbol{h}}, \quad s.t. \quad \sum\nolimits_i h_i = k, \tag{8}$$

where $\mathcal{B}^\tau = [0,1]^\tau$ is a unit box in $\mathbb{R}^\tau$.

Since problem (8) is a *linear-fractional programming* (LFP), it can be reduced to a linear programming problem of $\tau + 1$ variables and $\tau + 2$ constraints [2]. It follows that *exact* inference can be performed *efficiently* for the proposed latent variable classifier (2). The optimal solution is $\boldsymbol{h}^* = \boldsymbol{h}^*(k^*)$ where $k^* = \text{argmax}_k[f^*(\boldsymbol{v}; \boldsymbol{w}, k) + r(k)]$.

3.5. Learning

The learning problem is to determine the parameter vector $\boldsymbol{w}$, given a training set $\{\boldsymbol{v}_i, y_i\}_{i=1}^N$, where $y_i \in \mathcal{Y} = \{+1, -1\}$ indicates if the i-th sample belongs to the target event class c. This problem is identical to that of learning a *multiple-instance* (MI-SVM) [1] or a *latent* (L-SVM) [7] support vector machine (SVM).

A large margin predictor of form (2) is the solution of [1]

$$\begin{aligned} &\min_{\boldsymbol{w},\boldsymbol{\xi}} \ \frac{1}{2}||\boldsymbol{w}||^2 \ + \ C\sum\nolimits_{i=1}^N \xi_i \\ &\ s.t. \ \ y_i f_{\boldsymbol{w}}(\boldsymbol{v}_i) \geqslant 1 - \xi_i, \ \ \xi_i \geqslant 0, \quad \forall i \end{aligned} \tag{9}$$

This is a semi-convex problem, *i.e.*, a non-convex problem in general, which becomes convex if the latent variables are fixed for all *positive* examples. In this case, the objective function is quadratic and the feasible set is the intersection of a series of α-sublevel sets [2] of convex functions.

The solution of (9) is equivalent to that of the unconstrained problem

$$\min_{\boldsymbol{w}} \frac{1}{2}||\boldsymbol{w}||^2 + C\sum\nolimits_{i=1}^N \max\big(0, 1 - y_i f_{\boldsymbol{w}}(\boldsymbol{v}_i)\big), \tag{10}$$

for which a number of iterative algorithms have been proposed in the literature [1, 29, 7]. In this work, we adopt the *concave-convex procedure* (CCCP) of [29]. This consists of rewriting the objective of (10) as the sum of a convex and a concave functions

$$\begin{aligned} \min_{\boldsymbol{w}} \ &\Bigg[\frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i\in D_n} \max\big(0, 1 + f_{\boldsymbol{w}}(\boldsymbol{v}_i)\big) \\ &+ C\sum_{i\in D_p} \max\big(f_{\boldsymbol{w}}(\boldsymbol{v}_i), 1\big)\Bigg] + \Bigg[-C\sum_{i\in D_p} f_{\boldsymbol{w}}(\boldsymbol{v}_i)\Bigg], \end{aligned} \tag{11}$$

where D_p and D_n are the positive and negative training sets, respectively. CCCP then alternates between two steps.

The first computes a tight convex upper bound of the second (concave) term of (11), by estimating the configuration of hidden variables that best explains the positive training data under the current model. The second minimizes this upper bound, by solving a standard *structural SVM* [25] problem, which is convex, via either stochastic gradient descent [7], LIBLINEAR [6], cutting plane algorithms [25], or the proximal bundle method [12] (which we adopt in this work). The overall procedure resembles the popular *expectation-maximization* (EM) algorithm for estimation of the parameters of probabilistic models with latent variables.

4. Hypothesis Space for Pooled Features

In this section we discuss several possibilities for the hypothesis space of the proposed complex event detector.

4.1. Structure of the Pooling Window

The detector supports a number of possibilities with regards to the structure of $\boldsymbol{h}$. The first is *no selection, i.e.*, pooling from the entire sequence. This is equivalent to BoF with average pooling. The second is a *continuous window*, *i.e.*, the elements of $\boldsymbol{h}$ are all ones within a sliding continuous subset of the temporal locations: $h_i = 1$ if and only if $i \in \{t, \ldots, t+\delta\} \subset \{1, \ldots, \tau\}$. In this case, $\boldsymbol{h}$ is completely specified by a *window* (t, δ) with starting point t and duration δ. The use of such a sliding window provides a rough localization constraint for an activity, akin to the discriminative (continuous) subsequence of [18]. The third is a *temporally localized selector, i.e.*, an element of $\boldsymbol{h}$ can be one only inside the window (t, δ) but does not have to be active. The fourth is an *unconstrained selector* $\boldsymbol{h}$, which is a special temporally localized selector with window $(1, \tau)$. When a window (t, δ) is used, the starting point t is treated as an extra latent variable, whose optimal value is determined by repeating the inference of (2) at each window location and choosing the one with highest classification score. The duration δ is a parameter determined by cross-validation.

4.2. Structure of Pooled Features

So far, we have assumed that the features $\boldsymbol{x}_i$ of (1) are histogram of visual word counts of video segments $\boldsymbol{s}_i$. In fact, it is not necessary that the features $\boldsymbol{x}_i$ report to a single segment. While $\boldsymbol{\Phi}(\boldsymbol{v}, \boldsymbol{h})$ can pool, or average, single segment features $\boldsymbol{x}_i$, this may not be enough for discriminating certain types of events. Consider, for example, a traffic monitoring system confronted with two classes of events. The first consists of the sequence of atomic behaviors "car accelerates" and "car crashes", corresponding to regular traffic accidents. The second to a sequence of "car crashes" and "car accelerates", corresponding to accidents where one of the vehicles flees the accident site. In the absence of an explicit encoding of the temporal sequence of the atomic behaviors, the two events cannot be disambiguated. This problem has motivated interest in the detailed encoding of temporal structure [8, 27, 23, 14]. Some of these approaches are based on histograms of sophisticated spatio-temporal features, and could be integrated in the proposed detector. Another possibility is to extend the proposed pooling scheme to *tuples of pooling regions*. For example, dynamic pooling can be applied to *segment pairs*, by simply replacing the segment set $\mathcal{S}$ with

$$\mathcal{S}_2 = \{(\boldsymbol{s}_i, \boldsymbol{s}_j) | L_1 \leqslant j - i \leqslant L_2, \boldsymbol{s}_i, \boldsymbol{s}_j \in \mathcal{S}\} \subset \mathcal{S} \times \mathcal{S}, \quad (12)$$

where L_1 and L_2 are parameters that control the temporal distribution of the two segments. As shown in Figure 2, the feature of (1) naturally supports this representation. It suffices to make each column of X the concatenation of the features extracted from segment pairs $(\boldsymbol{s}_i, \boldsymbol{s}_j)$ in $\mathcal{S}_2$, with the latent variable $\boldsymbol{h}$ as an indicator of the selected pairs.

The procedure could be extended to η-tuples of higher order ($\eta > 2$) by concatenation of multiple segment features. The price to pay is computation complexity, since this increases the dimension of the hypothesis space from $O(2^\tau)$ to $O\left(2^{\tau^\eta}\right)$. In particular, (8) requires the solution of a linear program of $O(\tau^\eta)$ variables and constraints. In our experience, this is feasible even for large datasets when $\eta = 2$, *i.e.*, for segment pairs. We have not yet considered tuples of high order. It should be noted that the two-tuple extension generalizes some representations previously proposed in the literature. For example, when $L_1 = L_2 = 1$, the pair pooling strategy is similar to the *localized* version of the $t2$ temporal pyramid matching scheme of [13], albeit with dynamically selected pooling windows. $\mathcal{S}_2$ can also be seen as an automated two-tuple version of the representation of [8], where activities are manually decomposed into three atomic actions.

4.3. Learning with Different Pooling Strategies

The different possibilities for $\mathcal{H}$ can be explored synergistically during learning. This follows from the fact that, as happens to EM, CCCP is only guaranteed to converge to a local minima or saddle point [22]. Hence, a careful initialization is required to achieve good solutions. In our implementation, we rely on a four-step incremental refinement scheme to determine the initial solution. We start by learning a SVM without latent variables, *i.e.*, based on BoF. This is identical to [13] without temporal pyramid matching. It produces an SVM parameter $\boldsymbol{w}_{BoF}$ which is used to initialize the CCCP algorithm for learning an SVM with latent variables. In this second learning stage, the hidden variable selector $\boldsymbol{h}$ of (1) is restricted to a continuous pooling window (CPW), producing a latent SVM of parameter $\boldsymbol{w}_{CPW}$. This parameter is next used to initialize the CCCP algorithm for learning a latent SVM of temporally localized window for single segment pooling (SSP), *i.e.*, $\eta = 1$, with

Table 1: Average Precision for Activity Recognition on Olympic Sports Dataset.

Activity	BoF-TP [13]	DMS [17]	VD-HMM [23]	BDS [14]	Dynamic Pooling	
					SSP	SPP
high-jump	80.6%	68.9%	18.4%	**82.2%**	62.1%	69.1%
long-jump	86.0%	74.8%	81.8%	**92.5%**	74.4%	81.6%
triple-jump	51.5%	52.3%	16.1%	52.1%	44.6%	**54.3%**
pole-vault	60.9%	82.0%	**84.9%**	79.4%	59.7%	65.2%
gymnastics-vault	80.3%	**86.1%**	85.7%	83.4%	86.0%	85.0%
shot-put	39.6%	62.1%	43.3%	**70.3%**	60.8%	61.0%
snatch	58.8%	69.2%	88.6%	72.7%	65.1%	**89.7%**
clean-jerk	65.5%	84.1%	78.2%	85.1%	81.8%	**89.2%**
javelin throw	52.7%	74.6%	79.5%	**87.5%**	69.2%	79.9%
hammer throw	**81.7%**	77.5%	70.5%	74.0%	67.6%	72.3%
discus throw	40.4%	**58.5%**	48.9%	57.0%	47.9%	56.2%
diving-platform	**94.3%**	87.2%	93.7%	86.0%	89.2%	90.6%
diving-springboard	56.3%	77.2%	79.3%	78.3%	83.7%	**88.0%**
basketball-layup	69.8%	77.9%	85.5%	78.1%	83.3%	**86.1%**
bowling	61.7%	72.7%	64.3%	52.5%	**77.3%**	77.0%
tennis-serve	50.5%	49.1%	49.6%	38.7%	73.1%	**73.8%**
mean AP	64.4%	72.1%	66.8%	73.2%	70.4%	**76.2%**

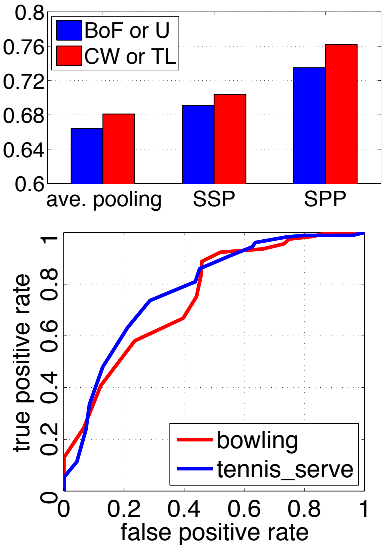

Figure 3: mAP of different pooling strategies and features on Olympic sports dataset (Top); and ROC curves for ground-truth sub-sequence detection by SSP for “bowling” and “tennis serve” on Olympic dataset (Bottom).

parameter $\boldsymbol{w}_{SSP}$. Finally, $\boldsymbol{w}_{SSP}$ is used to initialize CCCP for learning a latent SVM of temporally localized pooling window with segment pair selection (SSP), *i.e.*, $\eta = 2$.

5. Experiments

Several experiments were conducted to evaluate the performance of the proposed event detector, using three datasets and a number of benchmark methods for activity or event recognition. All these methods are based on BoF, obtained by detecting spatial-temporal interest points, extracting descriptors from these interest points, and quantizing these descriptors with a visual word dictionary learned from the training set [13, 28]. Unless otherwise specified, all experiments relied on the popular spatio-temporal interest point (STIP) descriptor of [13], and parameters of dynamic pooling were selected by cross-validation in the training set.

5.1. Olympic Sports

Olympic Sports [17] consists of around 50 sequences from each of 16 sports categories. While not really an open-source video collection (many of the sequences are extracted from sports broadcasts and depict a single well defined activity), this dataset is challenging for two main reasons: 1) some activities (*e.g.*, “tennis serve”, or “basketball layup”) have a variety of signature behaviors of variable location or duration, due to intra-class variability and poor segmentation/alignment; and 2) it contains pairs of confusing activities (*e.g.*, sub-types of a common category, such as the weight lifting activities of “snatch” and “clean-and-jerk”), whose discrimination requires fine-grained models of temporal structure. Low-level features were extracted from video segments of 30-frames (with an overlap of 15-frames) and quantized with a 4000-word codebook. Performance was measured with the mean per-category average precision (mAP), using 5-fold cross-validation.

Pooling Strategy We first evaluated the benefits of the various pooling structures of Section 4. The top of Figure 3 shows results for 4 structures: average pooling on the whole sequence (BoF), or on a continuous window (CW) (t, δ), temporally localized (TL) selector, and unconstrained (U) selector. The latter two were repeated for two feature configurations - single segments (SSP) and segment pairs (SPP) - for a total of 6 configurations. All dynamic pooling mechanisms outperformed BoF, with gains as high as 10%. In general, more adaptive pooling performed better, *e.g.*, CW better than BoF and TL better than CW. The only exception was the U selector which, while beating BoF and CW, underperformed its temporally localized counterpart (TL). This suggests that it is important to rely on a flexible selector $\boldsymbol{h}$, but it helps to localize the region from which segments are selected. With respect to features, pooling of segment pairs (SPP) substantially outperformed single segment pooling (SSP). This is intuitive, since the SPP representation accounts for long-term temporal video structure, which is important for the discrimination of similar activities (see discussion below). Given these observations, we adopted the TL pooling strategy in all remaining experiments.

Modeling Temporal Structure We next compared the proposed detector to prior methods for modeling the temporal structure of complex activities. The results are summarized in Table 1. BoF-TP had the worst performance. This was expected, given its coarse and static temporal pooling, which only works for categories with clear discriminant motion (*e.g.*, “diving-platform”). Methods that capture finer temporal structure, *e.g.*, decomposable motion segments (DMS) [17] (which decomposes an activity into six atomic behaviors temporally anchored at fixed video locations), and VD-HMM [23] or BDS [14] (which models the evolution of attribute sequences), performed better, sometimes beating TL-SSP; yet they were clearly outperformed

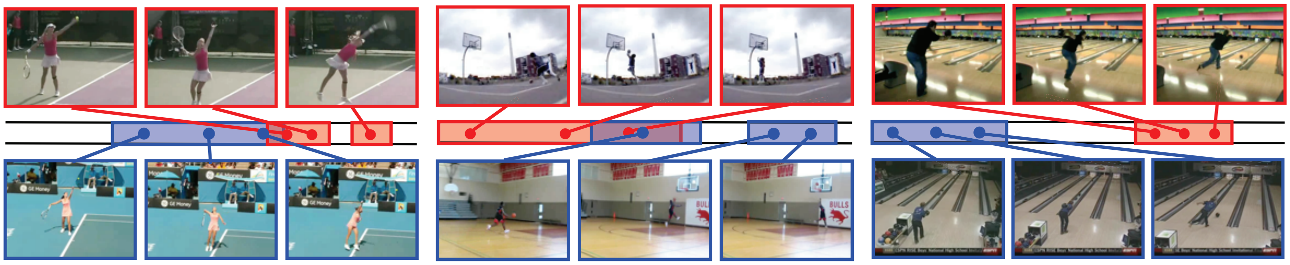

Figure 4: Characteristic segments (marked by shaded boxed region) of "tennis serve" (left), "basketball-layup" (middle) and "bowling" (right) discovered by SSP on Olympic. The bold black lines are normalized timelines of each sequence. Keyframes of the characteristic segments are shown with their anchor points in the timeline.

Table 2: mAP on Olympic Sports.

Method	Result
Wang *et al.* [27]	75.9%
Brendel *et al.* [3]	76.0%
Brendel & Todorovic [4]	77.3%
Gaidon *et al.* [9]	82.7%
Jiang *et al.* [11]	80.6%
Todorovic [24]	82.9%
SPP-SVM	**84.5%**

Table 3: Average Precision for Event Detection on TRECVID MED11 DEVT Dataset.

Event (E001-E005)	Random Guess	BoF-TP [13]	DMS [17]	VD-HMM [23]	BDS [14]	Dynamic Pooling	
						SSP	SPP
attempt a board trick	1.18%	16.47%	5.84%	15.44%	8.41%	18.18%	**26.09%**
feed an animal	1.06%	4.73%	2.28%	3.55%	1.78%	6.48%	**7.62%**
land a fish	0.89%	19.25%	9.18%	14.02%	6.20%	18.53%	**23.78%**
wedding ceremony	0.86%	32.17%	7.26%	15.09%	12.24%	**35.85%**	33.94%
work on a wood proj.	0.93%	20.59%	4.05%	8.17%	5.08%	**22.25%**	21.41%
mean AP	0.98%	18.64%	5.72%	11.25%	6.74%	20.26%	**22.57%**

by TL-SPP. This suggests that there are two important components of activity representation: 1) the selection of signature segments depicting characteristic behaviors; and 2) the temporal structure of these behaviors. Since most of sequences in this dataset are well segmented, the latter is more critical. TL-SSP, which only captures the location of signature segments, underperforms some of the previous models, which model the temporal structure. However, by not focusing on the segments of interest, the latter face too hard of a modeling challenge and are inferior to TL-SPP, which addresses both components. Note, in fact, that the prior models underperform even TL-SSP on categories with characteristic behaviors widely scattered across the video, *e.g.*, "bowling" and "tennis-serve". This is illustrated in Figure 4, which shows the segments selected by TL-SSP for the activities "tennis-serve", "basketball layup" and "bowling". Note that, despite the large variability of location of the characteristic behaviors in the video of these categories, *e.g.*, "throwing (ball)-waving (racket)-hitting (ball)" for "tennis-serve", TL-SSP is able to localize and crop them fairly precisely. This ability is also quantified in Figure 3 by a small experiment, where we 1) manually annotated the characteristic behaviors of "bowling" and "tennis-serve", and 2) compared this ground-truth to the video portion selected by TL-SSP. The resulting ROC curves clearly illustrate that performance of TL-SSP is better than chance.

State-of-the-Art The experiments above used STIP descriptors, for compatibility with other methods in Table 1. More recently, it has been shown that better performance is possible with dense trajectory feature (DTF) descriptors [27]. The best results on Olympic have been achieved with this descriptor [9, 11]. We have compared these benchmarks to an implementation of the proposed SPP-SVM that uses DTF, under the setting of [9]. As summarized in Table 2, SPP-SVM achieves the best results in the literature.

5.2. TRECVID-MED11

The second and third sets of experiments were conducted on the 2011 TRECVID multimedia event detection (MED) dataset [19]. It contains over $45,000$ videos of 15 high-level event classes (denoted "E001" to "E015") collected from a variety of Internet resources. The training set (denoted "EC"), contains 100 to 200 ground-truth instances of each event class, totaling over 2000 videos. The test set is split into two folds, denoted "DEVT" and "DEVO". The $10,403$ clips in DEVT contain positive samples from the classes E001 to E005 and negative samples that do not correspond to any of the 15 events. DEVO contains $32,061$ video clips of both positive and negative samples from classes E006 to E015. The large variation in temporal duration, scenes, illumination, cutting, resolution, *etc* in these video clips, together with the size of the negative class, make the detection task extremely difficult. In this dataset a $10,000$-word vocabulary is used and segments were 60-frame long with 30 frame overlapping. To improve discriminative power, we implemented the feature mapping of [26] for dynamic pooling and the baseline BoF-TP of [13].

Table 3 and Table 4 present the results of the different methods on the two datasets. Since, unlike Olympic, the videos are open-source, there is no pre-segmentation or alignment and plenty of irrelevant content. This is too much for approaches modeling holistic temporal structure like DMS [17], VD-HMM [23] and BDS [14], which significantly underperform the baseline BoF-TP. In these datasets,

Table 4: Average Precision for Event Detection on TRECVID MED11 DEVO Dataset.

Event (E006-E015)	Random Guess	BoF-TP [13]	DMS [17]	VD-HMM [23]	Dynamic Pooling	
					SSP	SPP
birthday party	0.54%	4.44%	2.25%	4.38%	**6.09%**	6.08%
change a veh. tire	0.35%	1.28%	0.76%	0.92%	1.90%	**3.96%**
flash mob gather.	0.42%	26.32%	8.30%	15.29%	31.19%	**35.28%**
get a veh. unstuck	0.26%	3.33%	1.95%	2.04%	4.54%	**8.45%**
groom an animal	0.25%	1.80%	0.74%	0.74%	**3.54%**	3.05%
make a sandwich	0.43%	**5.03%**	1.48%	0.84%	4.66%	4.95%
parade	0.58%	**9.13%**	2.65%	4.03%	8.72%	8.95%
parkour	0.32%	15.52%	2.05%	3.04%	17.86%	**24.62%**
repair an appliance	0.27%	16.62%	4.39%	10.88%	18.32%	**19.81%**
work on a sew. proj.	0.26%	5.47%	0.61%	5.48%	**7.43%**	6.53%
mean AP	0.37%	8.89%	2.52%	4.77%	10.52%	**12.27%**

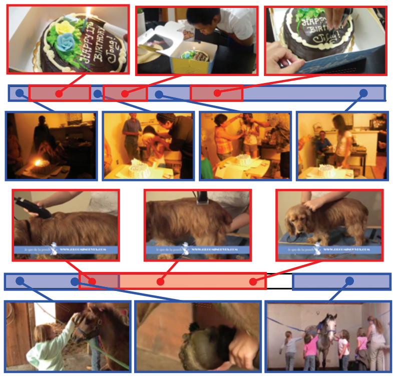

Figure 5: Signature segments discovered by SSP for "birthday party" (top) and "groom an animal" (bottom) on MED11.

both the identification of characteristic segments and the modeling of their temporal structure are important. Due to this, 1) both the SSP and SPP variants of the proposed detector outperform all other methods (note the large AP difference on events like "attempt a board trick", "feed an animal", *etc*), and 2) the gains of SPP over SSP are smaller than in Olympic, although still significant. Visual inspection indicates that SSP can provide quite informative content summarization of the video, as shown in Figure 5.

6. Conclusion

We proposed a joint framework for extracting characteristic behaviors, modeling temporal structure, and recognizing activity on video of complex events. It was shown that, under this formulation, efficient and exact inference for selection of signature video portion is possible over the combinatorial space of possible segment selections. An experimental comparison to various benchmarks for event detection, on challenging datasets, justified the effectiveness of the proposed approach.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *NIPS*, 2002. 4

[2] S. Boyd and L. Vandenberghe. *Convex optimization*. 2004. 4

[3] W. Brendel, A. Fern, and S. Todorovic. Probabilistic event logic for interval-based event recognition. *CVPR*, 2011. 7

[4] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. *ICCV*, 2011. 3, 7

[5] L. Cao, Y. Mu, N. Apostol, S.-F. Chang, G. Hua, and J. R. Smith. Scene aligned pooling for complex video recognition. In *ECCV*, 2012. 1, 2

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008. 5

[7] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 32(9):1627–1645, 2009. 4, 5

[8] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. *CVPR*, 2011. 3, 5

[9] A. Gaidon, Z. Harchaoui, and C. Schmid. Recognizing activities with cluster-trees of tracklets. *BMVC*, 2012. 2, 7

[10] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. *CVPR*, 2012. 2

[11] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. *ECCV*, 2012. 7

[12] K. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Program.*, 46:105–122, 1990. 5

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *CVPR*, 2008. 2, 3, 4, 5, 6, 7, 8

[14] W. Li and N. Vasconcelos. Recognizing activities by attribute dynamics. *NIPS*, 2012. 2, 3, 5, 6, 7

[15] W. Li and N. Vasconcelos. Exact linear relaxation of integer linear fractional programming with non-negative denominators. *SVCL Technical Report*, 2013. 4

[16] W. Li, Q. Yu, H. Sawhney, and N. Vasconcelos. Recognizing activities via bag of words for attribute dynamics. *CVPR*, 2013. 1, 3

[17] J. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. *ECCV*, 2010. 1, 2, 3, 6, 7, 8

[18] S. Nowozin, G. Bakir, and K. Tsuda. Discriminative subsequence mining for action classification. *ICCV*, 2007. 2, 5

[19] P. Over, G. Awad, J. Fiscus, B. Antonishek, M. Michel, A. F. Smeaton, and W. Kraaij. Trecvid 2011 – an overview of the goals, tasks, data, evaluation mechanisms, and metrics. *Proceedings of TRECVID 2011*, 2011. 7

[20] S. Satkin and M. Hebert. Modeling the temporal extent of actions. *ECCV*, 2010. 2, 3

[21] K. Schindler and L. V. Gool. Action snippets: How many frames does human action recognition require? *CVPR*, 2008. 2

[22] B. K. Sriperumbudur and G. R. G. Lanckriet. A proof of convergence of the concave-convex procedure using zangwill's theory. *Neural Computation*, 24:1391–1407, 2012. 5

[23] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. *CVPR*, 2012. 1, 2, 3, 5, 6, 7, 8

[24] S. Todorovic. Human activities as stochastic kronecker graphs. *ECCV*, 2012. 3, 7

[25] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005. 5

[26] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE TPAMI*, 34(3):480–492, 2012. 3, 7

[27] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. *CVPR*, 2011. 2, 5, 7

[28] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. *BMVC*, 2009. 3, 6

[29] A. L. Yuille and A. Rangarajan. The concave-convex procedure (cccp). *NIPS*, 2003. 4