Recognizing Complex Events in Videos by Learning Key Static-Dynamic Evidences

Kuan-Ting Lai[1,2], Dong Liu[3], Ming-Syan Chen[1,2], and Shih-Fu Chang[3]

[1] Graduate Institute of Electrical Engineering, National Taiwan University, Taiwan
[2] Research Center for IT Innovation, Academia Sinica, Taiwan
[3] Department of Electrical Engineering, Columbia University, USA
{ktlai,mschen}@arbor.ee.ntu.edu.tw, {dongliu,sfchang}@ee.columbia.edu

Abstract. Complex events consist of various human interactions with different objects in diverse environments. The evidences needed to recognize events may occur in short time periods with variable lengths and can happen anywhere in a video. This fact prevents conventional machine learning algorithms from effectively recognizing the events. In this paper, we propose a novel method that can automatically identify the key evidences in videos for detecting complex events. Both static instances (objects) and dynamic instances (actions) are considered by sampling frames and temporal segments respectively. To compare the characteristic power of heterogeneous instances, we embed static and dynamic instances into a multiple instance learning framework via instance similarity measures, and cast the problem as an Evidence Selective Ranking (ESR) process. We impose ℓ_1 norm to select key evidences while using the Infinite Push Loss Function to enforce positive videos to have higher detection scores than negative videos. The Alternating Direction Method of Multipliers (ADMM) algorithm is used to solve the optimization problem. Experiments on large-scale video datasets show that our method can improve the detection accuracy while providing the unique capability in discovering key evidences of each complex event.

Keywords: Video Event Detection, Infinite Push, Key Evidence Selection, ADMM.

1 Introduction

Recognizing complex multimedia event in videos is becoming increasingly important in the field of computer vision. In 2010, the TREC Video Retrieval Evaluation (TRECVID) [15] Multimedia Event Detection (MED) evaluation task defined a wide range of complex events, and spurred broad research interests in the computer vision community. These complex events include "attempting board trick", "landing a fish", "changing a vehicle tire", and "flash mob gathering", to name a few. In contrast to the human activity videos in action recognition [19], which mainly focus on a single person's simple motions in the 5 to 10 seconds short video clips, the complex event videos consist of various interactions of human actions and objects in different scenes, and may last from

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 675–688, 2014.

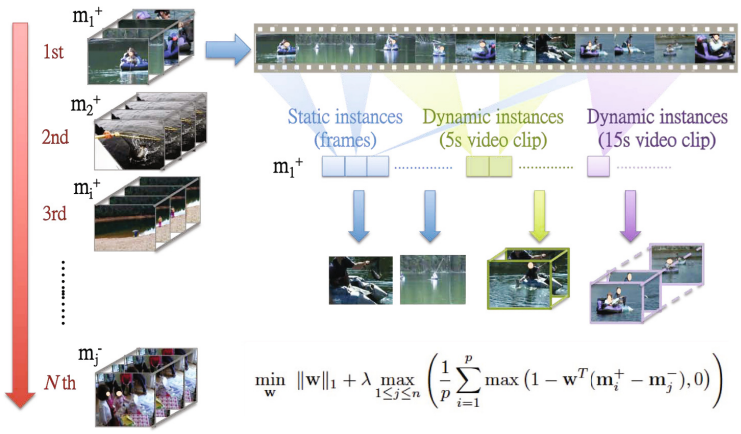

Fig. 1. The proposed Evidence Selective Ranking (ESR) framework. The static/dynamic instances of variable lengths are first extracted from a video, and mapped to a pre-learnt static/dynamic instance codebook via maximum similarity measure (instance embedding). The heterogeneous embedded vectors are then concatenated and trained by Infinite Push loss function with ℓ_1 norm to select the key evidences while enforcing positive videos to have higher detection scores.

several minutes to even an hour. Therefore, it is challenging to develop robust event detection models that can precisely capture the essential information in event videos.

Although many algorithms have been proposed to recognize complex events [8], the most popular method is still aggregating the raw audio/visual/motion features extracted from the videos into different variants of Bag-of-Words (BoW) histogram, and then feed it into sophisticated statistical learning models for event modeling. However, the main issue with this strategy is that it treats different components of a long event video as equally important, and ignores the fact that an event video may contain significant amount of background components that have no direct association with the target event. In fact, a complex event can usually be recognized by spotting a few key static and/or dynamic evidences [2]. For example, a "wedding ceremony" video can be correctly detected by successfully identifying several static frames containing *bride* and *groom*, while a "attempting bike trick" video can be detected by spotting some dynamic short segments containing the activity of *jumping with a bike*.

This motivates us to develop a method that is able to identify the key static-dynamic evidences in event videos and leverage them for improving the overall performance of event detection. Nevertheless, this is a nontrivial task due to the following reasons. First, given a complex event video, there are large amounts of frames and video segments that can be potential evidences, and the characteristic power of heterogeneous instances cannot be directly compared. To address those issues, we employ the instance embedding method [4] to map different kinds of instances into pre-learnt instance codebooks and concatenate the

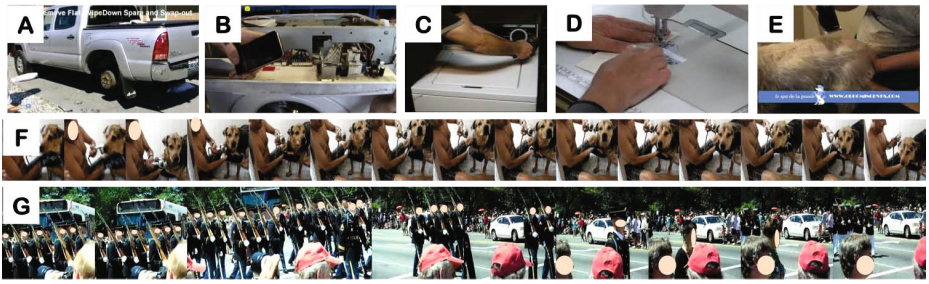

Fig. 2. Some of the top key event evidences selected by our method. (A) to (E) are static instances, while (F), (G) are dynamic instances with 15 and 10 seconds length. (A) is changing a tire; (B), (C) are fixing an appliance; (D) is sewing project; (E), (F) are grooming animals; (G) is parade.

embedded vectors. Specifically, given an event video set, we first sample frames from each video as static instances and short video segments at varied length as the dynamic instances. The static and dynamic instances are then clustered respectively to form the static and dynamic instance codebooks. Finally, we map all static/dynamic instances in a video onto the static/dynamic instance codebook, in which the value of each static/dynamic codeword is determined by the maximal similarity between all static/dynamic instances and the codeword. In this way, we end up with a compact heterogeneous instance representation that comprehensively encodes static and dynamic instances in each video.

Second, even after we have a compact instance representation, we need to investigate novel solutions that can select most distinctive evidences (positive instances) from videos and effectively utilize the information to detect complex events. Indeed, the video event detection task can be seen as a ranking process that aims at assigning higher detection scores to positive videos than negative videos. This inspires us to formulate event detection problem as an Evidence Selective Ranking (ESR) procedure, which discovers the key static-dynamic evidences in event videos while directly enforcing positive videos to have the highest scores in the detection results. Specifically, a ℓ_1-norm is first imposed to induce sparsity on the heterogeneous instance representation and determine a subset of dimensions. To ensure that the positive videos have the highest detection scores, we use $\ell_{1,\infty}$ infinite push loss to maximize the number of positive videos having higher detection scores than the negative videos. With this evidence selective ranking process, we can identify the key static-dynamic evidences while pushing the positive videos to rank at higher positions in the ranking list of detection result. Figure 1 illustrates the framework of our proposed method.

In the following sections, we will demonstrate experimentally that the proposed ESR method can achieve significant performance gains over various video event detection benchmarks. We will also show that our method is able to reveal the key static-dynamic evidences for identifying a video event (see Figure 2).

2 Related Work

Complex event detection has attracted many research interests in recent years. A recent literature review can be found in [8]. A video event detection system usually consists of the following procedures: feature extraction, quantization/pooling, training/recognition, and multimodal fusion. The local low-level features include static features, spatio-temporal features and audio features. Recently the Dense Trajectory based Features (DTF) [24] achieved great results on action recognition and is widely applied in event detection system. In terms of training/recognition approaches, the current methods can be roughly categorized into large margin based methods, graphical models, and knowledge based techniques. The commonly used method is baesd on large margin framework with kernel techniques. Most previous methods represent video as an aggregated global feature vector and train the event model with SVM [8,12,20]. However, as aforementioned, these approaches treat all evidences in videos as equally important and cannot effectively leverage the key evidences to improve the detection performance. To alleviate the above issue, some existing works exploited the short segments in event videos to improve event detection performance. Cao *et al.* [3] proposed a scene aligned pooling method for video representation. The basic assumption in this method is that a video clip is often composed of segments of different scenes, and this motivates the authors to perform video feature pooling within the individual scenes. However, the main focus of this work is to obtain a robust video feature pooling result, and cannot judiciously select the key evidences in event videos as our method does. Similarly, Li *et al* [10] proposed a Dynamic Pooling method for event detection, in which an event video is decomposed into short segments, and the most informative segments for detecting this event are identified through latent variable inference and used for video feature pooling. Differently, our method focuses on selecting the most informative evidences in videos, which goes beyond feature pooling procedure and achieves better performance than the method in [10] (see Table 2).

One available solution for learning key evidences in videos is Multiple Instance Learning (MIL). Initially MIL was introduced to solve drug design problem [5]. The labels are given to bags (drugs) instead of to the instances (molecules) inside. A bag is labeled as positive if at least one of its instance is positive, or negative if all its instances are negatives. This assumption works well for drug design because only one molecule form works for a drug. But in computer vision applications, the positive and negative bags may share some visual cues in common, and the above assumption is typically not true. In contrary, our method based on instance embedding [4,6] does not make any assumption on the instances in videos, and directly chooses any number of the most useful instances for event modeling.

Methodologically, our method adopts learning-to-rank algorithm to perform video event detection. One classic large-margin ranking algorithm that can be applied is Ranking SVM [9]. However, it focuses on optimizing pairwise ranking accuracy without considering the entire ranking list. Newly developed ranking algorithms, such as p-norm push [18] and Infinite Push [16], put emphasis on

optimizing the accuracy at the top of the rank list, which is more suitable for event detection. Inspired by the Infinite Push ranking [1], which is the generalization bound of l_p norm push, we utilize the infinite push model to ensure a good ranking in the video detection results.

3 Evidence Selective Ranking for Video Event Detection

3.1 Compact Heterogeneous Instance Representation

Suppose there is an event video collection $\mathcal{X} = \{X_i\}_{i=1}^N$ with N videos, where $X_i = \{S_i \bigcup D_i\}$ is a video consisting of a static instance subset $S_i = \{\mathbf{s}_{i1}, \ldots, \mathbf{s}_{i,n_i}\}$ with n_i static frames and a dynamic instance subset $D_i = \{\mathbf{d}_{i1}, \mathbf{d}_{i2}, \ldots, \mathbf{d}_{i,m_i}\}$ with m_i dynamic segments. Here $\mathbf{s}_{ij} \in \mathbb{R}^{k_s}$ and $\mathbf{d}_{ij} \in \mathbb{R}^{k_d}$ are respectively the feature vector of the j-th static and dynamic instance of video X_i with k_s and k_d being the feature dimensionality. Furthermore, we collect all frames and segments into a static instance set $\mathcal{S} = \{S_i\}_{i=1}^N$ and a dynamic instance set $\mathcal{D} = \{D_i\}_{i=1}^N$.

We first construct codebooks for the static and dynamic instance set respectively. Specifically, we perform K-means clustering to partition $\mathcal{S}$ and $\mathcal{D}$ into G_s and G_d clusters, and treat each cluster center as one codeword. We define $\mathcal{V}_s = \{\mathbf{c}_1^s, \ldots, \mathbf{c}_{G_s}^s\}$ and $\mathcal{V}_d = \{\mathbf{c}_1^d, \ldots, \mathbf{c}_{G_d}^d\}$ as the static and dynamic codebooks, where $\mathbf{c}_i^s \in \mathbb{R}^{k_s}$ ($\mathbf{c}_i^d \in \mathbb{R}^{k_d}$) is the i-th codeword in static (dynamic) codebook.

Next, the static and dynamic instances in a video are mapped onto their respective codebooks to generate the heterogeneous instance representation. In this work, we apply a similarity embedding method in [4] to effectively encode multiple instances in a video onto each codeword. Given the static instance set S_i of video X_i, its encoding value on the l-th static codeword $\mathbf{c}_l^s$ is defined as:

$$s(S_i, \mathbf{c}_l^s) = \max_{1 \le j \le n_i} \exp\left(-\frac{d(\mathbf{s}_{ij}, \mathbf{c}_l^s)}{\sigma}\right), \tag{1}$$

where $d(\mathbf{s}_{ij}, \mathbf{c}_l^s)$ is the χ^2 distance function which measures the distance between an instance $\mathbf{s}_{ij}$ and codeword $\mathbf{c}_l^s$. σ is the radius parameter of the Gaussian function, which is set as the mean value of all pairwise distances among the static instances. The encoding value of the dynamic instance set D_i of video X_i can be calculated in a similar way. In the end, video X_i is encoded as a compact static-dynamic instance vector $\mathbf{m}_i \in \mathbb{R}^{G_s+G_d}$:

$$\mathbf{m}_i = [s(S_i, \mathbf{c}_1^s), \ldots, s(S_i, \mathbf{c}_{G_s}^s), s(D_i, \mathbf{c}_1^d), \ldots, s(D_i, \mathbf{c}_{G_d}^d)]^\top. \tag{2}$$

In the heterogeneous instance representation, each codeword in static/dynamic codebook characterizes a consistent static/dynamic pattern. When mapping the static/dynamic instances in a video onto one codeword, we use the maximum similarity to choose the most similar instance to generate the encoding value. This essentially measures the maximal coherence between the instances in a video and one pattern in the entire video set, and thus achieves robust heterogeneous instance representation.

3.2 Evidence Selective Ranking

Given an event category, assume we have a labeled training video set $\{\mathbf{m}_i, y_i\}_{i=1}^{V}$ with V videos, in which $\mathbf{m}_i$ is the static-dynamic evidence vector of the i-th video, and $y_i \in \{0, 1\}$ is the event label. To ease the following presentation, we partition all labeled training videos into a positive subset $\mathcal{P} = \{\mathbf{m}_i^+\}_{i=1}^{p}$ and a negative subset $\mathcal{N} = \{\mathbf{m}_i^-\}_{i=1}^{n}$, where $\mathbf{m}_i^+$ and $\mathbf{m}_i^-$ denote the evidence vector of a positive video and a negative video. p and n are respectively the total number of positive and negative training videos.

We want to learn an event detection function $f(\mathbf{m}) = \mathbf{w}^\top \mathbf{m}$, where $\mathbf{w} \in \mathbb{R}^{G_s+G_k}$ is the parameter vector. Our evidence selective ranking based event detection method is formulated as follows:

$$\min_{\mathbf{w}} \quad \|\mathbf{w}\|_1 + \lambda \ell(\mathcal{P}, \mathcal{N}; \mathbf{w}), \tag{3}$$

where λ is a tradeoff parameter among the two terms. The first term is a ℓ_1 norm induced sparse regularization on the heterogeneous instance representation that explicitly selects a subset of codeword dimensions. Such selected dimensions can be used to identify the key evidences in each event video. Specifically, given a selected dimension, the corresponding key evidence in a video is actually the instance that has been used to generate the encoding value on this dimension (i.e., the one which has maximal similarity with the corresponding codeword of this given dimension).

The second term is a ranking loss function, which is used to penalize a misranked pair in which the negative video has higher detection score than the positive one. In principle, we can instantiate this loss with any loss function in the learning-to-rank algorithms. In this work, we choose the recently introduced Infinite Push loss function as the loss function in our model due to its outstanding performance [1]. The objective of Infinite Push is to maximize the number of positive videos on the absolute top positions of the entire video rank list, without paying too much attention about getting an accurate ranking order among other parts of the list, which perfectly matches the goal of video event detection.

To design the Infinite Push loss function, the authors notice that maximizing positive videos at top is equivalent to minimize the number of positive videos scored lower than the highest-scored negative video. Furthermore, the number of positive videos scored lower than the highest-scored negative video is equivalent to the largest number of positive training videos scored lower than any negative video, which is a fraction of the total number of positive videos p and can be defined as:

$$\ell(\mathcal{P}, \mathcal{N}; \mathbf{w}) = \max_{1 \le j \le n} \left(\frac{1}{p} \sum_{i=1}^{p} I_{\mathbf{w}^T \mathbf{m}_i^+ < \mathbf{w}^T \mathbf{m}_j^-} \right), \tag{4}$$

where $I_{(\cdot)}$ is the indicator function which is 1 if the argument is true or 0 otherwise. Directly optimizing Eq. (4) is infeasible due to its discrete nature. Therefore, it is relaxed into a convex upper bound as below:

$$\ell(\mathcal{P}, \mathcal{N}; \mathbf{w}) = \max_{1 \le j \le n} \left(\frac{1}{p} \sum_{i=1}^{p} \max\left(1 - \mathbf{w}^\top (\mathbf{m}_i^+ - \mathbf{m}_j^-), 0\right) \right), \tag{5}$$

Based on the above definition, the objective function can be rewritten as:

$$\min_{\mathbf{w}} \ \|\mathbf{w}\|_1 + \lambda \max_{1\le j\le n} \left(\frac{1}{p} \sum_{i=1}^{p} \max\left(1 - \mathbf{w}^T(\mathbf{m}_i^+ - \mathbf{m}_j^-), 0\right) \right), \tag{6}$$

The above objective function is actually the sparse support vector Infinite Push recently proposed in [17], which is convex and thus can achieve global optimum. In the next subsection, we will elaborate on the optimization procedure.

3.3 Optimization Procedure

We directly adopt the Alternating Direction Method of Multipliers (ADMM) iterative optimization procedure in [17] to solve the optimization problem. The objective function is first rewritten as the following linearly-constrained problem:

$$\begin{aligned} \min_{\mathbf{w},\{a_{ij}\}} \quad & \|\mathbf{w}\|_1 + \lambda \max_{1\le j\le n} \left(\frac{1}{p} \sum_{i=1}^{p} \max(a_{ij}, 0) \right), \\ \text{s.t.,} \quad & a_{i,j} = 1 - \mathbf{w}^\top(\mathbf{m}_i^+ - \mathbf{m}_j^-). \end{aligned} \tag{7}$$

By defining matrix $\mathbf{M}$ whose rows are of the form $(\mathbf{m}_i^+ - \mathbf{m}_j^-)^\top$, vector $\mathbf{a}$ composing of all α_{ij}'s and function $g(\mathbf{a}) = \lambda \max_j(\frac{1}{p}\sum_i \max(a_{ij}, 0))$, the optimization can be rewritten as :

$$\begin{aligned} \min_{\mathbf{w},\mathbf{a}} \ & \|\mathbf{w}\|_1 + g(\mathbf{a}), \\ \text{s.t.,} \ & \mathbf{M}\mathbf{w} + \mathbf{a} - 1 = 0. \end{aligned} \tag{8}$$

The augmented Lagrangian of the above problem is:

$$\mathcal{L}(\mathbf{w}, \mathbf{a}, \delta, \mu) = \|\mathbf{w}\|_1 + g(\mathbf{a}) + \delta^\top(\mathbf{M}\mathbf{w} + \mathbf{a} - \mathbf{1}) + \frac{\mu}{2}\|\mathbf{M}\mathbf{w} + \mathbf{a} - \mathbf{1}\|^2, \tag{9}$$

where δ is a vector of Lagrangian multipliers for the equality constraint, and μ is a parameter of quadratic penalty setting as 10^{-4} according to the suggestion in ADMM procedure. The formula can be rearranged as:

$$\mathcal{L}(\mathbf{w}, \mathbf{a}, \gamma) = \|\mathbf{w}\|_1 + g(\mathbf{a}) + \frac{\mu}{2}\|\mathbf{M}\mathbf{w} + \mathbf{a} - \mathbf{1} + \gamma\|^2, \tag{10}$$

where $\gamma = \frac{\delta}{\mu}$. Finally, the problem can be solved alternatively at iteration k the following subproblems:

$$\mathbf{w}^{k+1} = \arg\min_{w} \mathcal{L}(\mathbf{w}, \mathbf{a}^k, \gamma^k), \tag{11}$$

$$\mathbf{a}^{k+1} = \arg\min_{a} \mathcal{L}(\mathbf{w}^{k+1}, \mathbf{a}, \gamma^k), \tag{12}$$

$$\gamma^{k+1} = \gamma^k + \mathbf{M}\mathbf{w} + \mathbf{a} - \mathbf{1}. \tag{13}$$

In particular, subproblem in Eq. (11) can be solved as a standard Lasso problem. Subproblem in Eq. (12) can be solved by first decoupling $\mathbf{a}$ into $\mathbf{a}^+$ and $\mathbf{a}^-$, and then solving them by Block Coordinate Descent as introduced in [17]. ADMM has a fast convergence rate of $O(1/t)$, where t is the iteration number. The running time of ESR will be reported in our experiments.

4 Experiments

In this section, we will evaluate the effectiveness of our Evidence Selective Ranking (ESR) method over the currently largest video datasets: TRECVID Multimedia Event Detection (MED) 2011 and 2012. In MED evaluation tasks, the test events of each year include events from pervious years. There are 15 events in MED 2011 and 25 events in MED 2012, which are listed in Table 1. We compare our ESR method with (1) Static instance (ST-inst) only. (2) Dynamic instances (Dyn-inst) only. (3) MILES [4], which is based on instance embedding and ℓ_1 SVM feature selection. We train an event model with a binary SVM classifier after the features are selected, and (4) The state-of-the-art event detection methods.

To generate the static instances, we extract frames from each video every 2 seconds and scale them down to 320×240 pixels. Then the SIFT features [11] are extracted by dense SIFT function in VLFeat library [23] with a 10-pixel step. Finally, each frame is represented as a $5,000$-dimensional SIFT BoW. The dynamic instances are generated by applying the sliding window approach to each video clip. We consider 5 kinds of video segments with different lengths as all dynamic instances in a video, in which we adopt 3, 5, 10, 15, 20 seconds sliding windows with 2, 3, 7, 10, 15 seconds overlapping to extract segments. For static and each of the 5 dynamic instances, the Yael K-means library [7] is used to learn a codebook with $5,000$ codewords. The final static-dynamic video instance vector is the concatenation of all encoding values over all 6 kinds of codebooks, which has $30,000$ feature dimensions in total.

To evaluate the performance of each method, the Average Precision (AP) is employed as the evaluation metric. Regarding the parameter setting, we use

Table 1. The 25 events defined in TRECVID MED 2011 and 2012

ID	MED 2011 Events	ID	MED 2012 Events
1	Attempting board trick	16	Attempting bike trick
2	Feeding animals	17	Cleaning appliance
3	Landing a fish	18	Dog show
4	Wedding ceremony	19	Give directions to location
5	Woodworking project	20	Marriage proposal
6	Birthday party	21	Renovating a home
7	Changing a tire	22	Rock climbing
8	Flash mob gathering	23	Town hall meeting
9	Getting vehicle unstuck	24	Win race without a vehicle
10	Grooming animal	25	Work on metal craft project
11	Making sandwich		
12	Parade		
13	Parkour		
14	Repairing appliance		
15	Work on sewing project		

3-fold cross-validation and vary the value of parameter $\lambda = \{0.1, 1, 10\}$ in the objective function to determine the appropriate parameter for each method.

4.1 Experiment on TRECVID MED 2011

The official MED 2011 dataset consists of three data splits: Event Collection (EC), the development collection (DEVT) and test collection (DEVO). The EC set contains $2,680$ training videos over 15 events. The DEVT set includes $10,403$ videos and is provided for participants to validate their systems. The DEVO set containing $32,061$ test videos is used to evaluate final performance. In MED 2011, the length of the videos ranges from several seconds to one hour. In this experiment, we follow these official data splits, in which we use EC and DEVT set to train/validate and use DEVO set to test. Notice that DEVO set does not include any videos of Event 1 to Event 5, so only test results of Event 6 to Event 15 are reported. Empirically we can achieve satisfactory results within only 5 iterations, so we set the max iterations of our ESR to 5 to save running time. The average running time of ESR for each MED11 event on a single Intel Xeon 2.67GHz core is around one hour.

Figure 4 and Table 2 show the performance of different methods in comparison, in which Table 2 mainly quotes the state-of-the-art results in literature. These results are from the recent proposed methods including DMS [13], VD-HMM [21], dynamic pooling with segment-pairs (SPP) [10] and multiple kernel latent SVM (MKL-KLSVM) [22], each of which follows the same setting of the official MED 11 data splits.

Table 2. The APs of different methods on TRECVID MED11 DEVO dataset

Event Name (006 - 015)	DMS [13]	VD-HMM [21]	SPP [10]	MKL-KLSVM[22]	MILES (SVM-$l1$)	Our Method
Birthday party	2.25%	4.38%	6.08%	6.24%	5.08%	**7.45%**
Change a vehicle tire	0.76%	0.92%	3.96%	**24.62%**	9.50%	14.44%
Flash mob gathering	8.30%	15.29%	35.28%	37.46%	33.77%	**40.87%**
Get a vehicle. unstuck	1.95%	2.04%	8.45%	**15.72%**	7.38%	7.72%
Groom an animal	0.74%	0.74%	**3.05%**	2.09%	1.76%	1.83%
Make a sandwich	1.48%	0.84%	4.95%	**7.65%**	3.13%	4.86%
Parade	2.65%	4.03%	8.95%	12.01%	14.34%	**17.69%**
Parkour	2.05%	3.04%	24.62%	10.96%	20.14%	**25.3%**
Repair an appliance	4.39%	10.88%	19.81%	**32.67%**	25.81%	31.75%
Work on sewing project	0.61%	5.48%	6.53%	7.49%	4.66%	**8.34%**
mean AP	2.52%	4.77%	12.27%	15.69%	12.56%	**16.02%**

From the results, we have the following observations: (1) The proposed ESR method produces better results than all other methods in comparison, which demonstrates its effectiveness in the task of video event detection. (2) The ESR method performs significantly better than the single instance based methods,

and (3) Our ESR method shows performance improvement over MILES method that detects videos based on SVM classifier instead of the infinite push ranking model. This verifies the benefits of introducing ranking model to event detection task. Notice that some new proposed technique like Fisher vector can be adopted in our framework and further improve the recognition accuracy [14].

As mentioned, one advantage of our method is that it is capable of locating the selected key evidences in each video for further visualization and analysis. Recall that, although the instances in a video are embedded into instance codebook space, the selected instances can be located simply by searching the instances with maximum similarities to the instance codebook in a video. Using this method, Figure 3 shows three of top evidences with the largest weights in the videos of some exemplary events, in which the static evidence is represented as a frame and the dynamic evidence is represented as a sequence of successive frames in the selected key segment. As can be seen, for "flash mob gathering" and "parade", the most distinctive event evidences are dynamic instances; for "change vehicle tire", "fix an appliance", and "sewing project", the selected event evidences are mainly static frames. These selected evidences are interpretable for human and useful for analyzing event videos.

To study the influences of the length of the video segments, we generate video segments with length of 3, 5, 10, 15 and 20, and use each kind of segments as dynamic instances to run our ESR method. We compare these results with our proposal that mixes segments of different lengths together, and the results can be shown in Figure 5. From the results, we can see that when there is only one fixed length dynamic instance for evidence selection, the 3-second and 5-second short video segments achieve best results on most events. However, using mixed length segments as dynamic instances always generates better performance than others, which confirms the soundness of our proposed dynamic evidence generation strategy.

In Figure 6, we further plot the proportions of each kind of evidences (both static and dynamic) selected from training videos of each MED11 event. As shown, event "flash mob gathering", "getting vehicle unstuck", and "grooming animals" have higher proportion of dynamic evidences, while "birthday party", "changing a tire" and "working on sewing project" have selected more static evidences. The evidence proportion distributions are intuitive to human and further show the advantages of our method.

4.2 TRECVID MED 2012

The MED12 dataset contains 25 complex events as shown in Table 1, which includes 15 events in MED11. The total training videos of the 25 events is $5,816$ videos. We choose two thirds of the data as training set ($3,878$ videos) and use the rest as test set ($1,938$ videos). In this experiment, we follow the same setting as we did in MED11. The APs of MED12 events are shown in Figure 7. The average running time of ESR is similar to MED11 events since the dimensions of heterogenous instance vectors are the same. Once again, the experiment results confirm the effectiveness of our proposed event detection method.

(a) change a vehicle tire

(b) flash mob gathering

(c) parade

(d) fix an appliance

(e) sewing project

Fig. 3. The top static/dynamic evidences selected for identifying target events

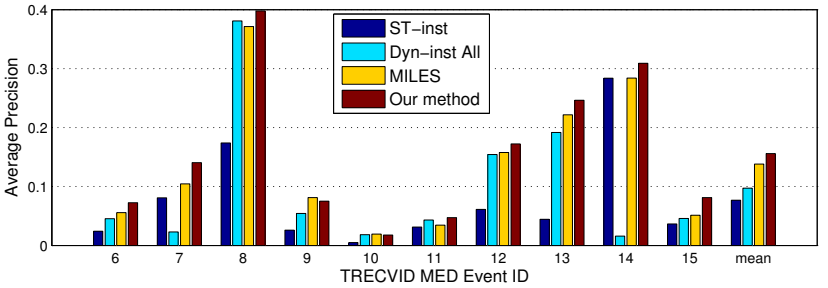

Fig. 4. The APs of different methods over TRECVID MED11 dataset. The methods in comparison include static instance only (ST-inst), all dynamic instances (Dyn-inst all), MILES with ℓ_1 SVM based feature selection and our ESR method.

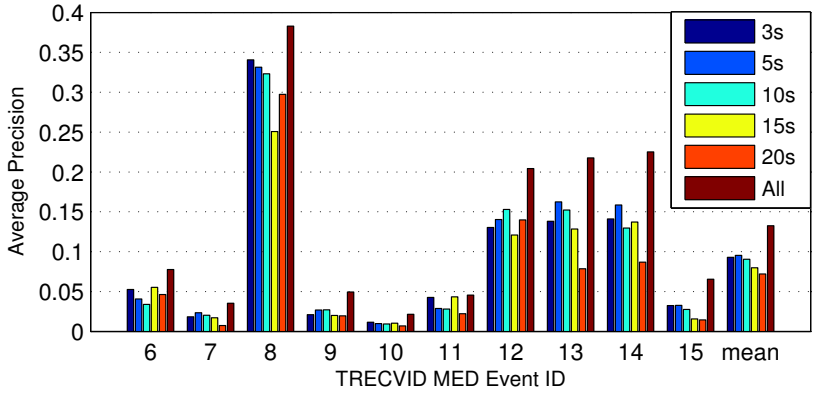

Fig. 5. The APs of dynamic instances with varied time lengths (3, 5, 7, 15, 20 seconds) on TRCVID MED11 DEVO dataset. "All" represents the result of using all kinds of dynamic instances. The applied low-level feature is MBH [25].

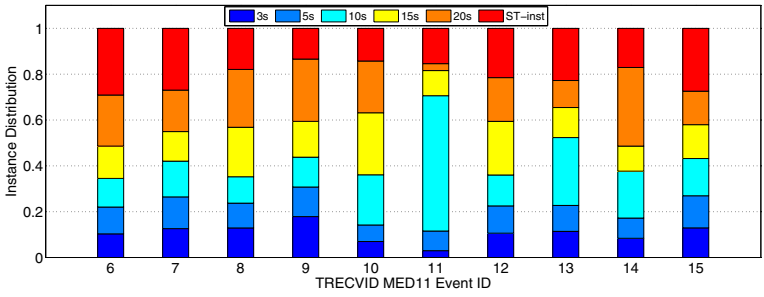

Fig. 6. The distribution of selected key evidences for MED11 events. The region in red represents the proportion of static instance (ST-inst), while others represent dynamic instance with different time length.

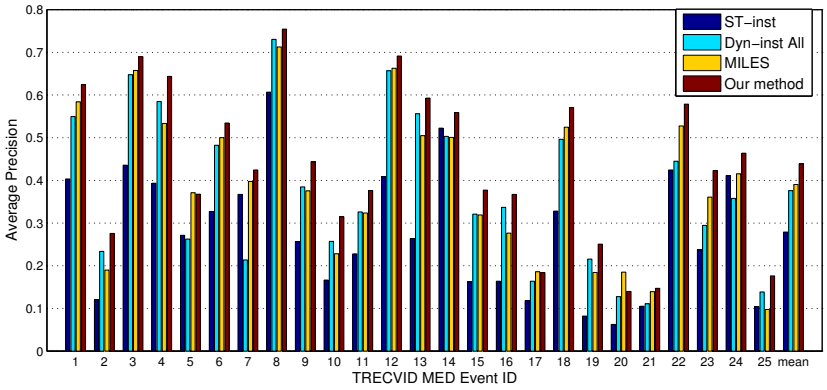

Fig. 7. The APs of different methods over TRECVID MED12 dataset. The methods in comparison include static instance only (ST-inst), all dynamic instances (Dyn-inst all), MILES with ℓ_1 SVM based feature selection and our ESR method.

5 Conclusion

We have proposed a novel event detection method by selecting key static-dynamic evidences from video content. To represent the static and dynamic evidences in videos, we encode the static frames and dynamic video segments into a compact heterogeneous instance representation through codebook generation and similarity mapping. Then a novel Infinite Push Ranking algorithm with ℓ_1-norm regularization is applied to simultaneously select the most useful evidences and rank positive videos at the top positions in the event detection rank list. Furthermore, the evidences discovered in our framework are interpretable for human, which can facilitate deep analysis of the complex events. The experimental results on large video dataset are promising and verify the effectiveness of our method.

References

1. Agarwal, S.: The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list. In: SDM, pp. 839–850. Society for Industrial and Applied Mathematics (2011)
2. Bhattacharya, S., Yu, F.X., Chang, S.F.: Minimally needed evidence for complex event recognition in unconstrained videos. In: ICMR (2014)
3. Cao, L., Mu, Y., Natsev, A., Chang, S.-F., Hua, G., Smith, J.R.: Scene aligned pooling for complex video recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 688–701. Springer, Heidelberg (2012)
4. Chen, Y., Bi, J., Wang, J.Z.: Miles: Multiple-instance learning via embedded instance selection. PAMI 28(12), 1931–1947 (2006)
5. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence 89(1), 31–71 (1997)

6. Ikizler-Cinbis, N., Sclaroff, S.: Object, scene and actions: Combining multiple features for human action recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 494–507. Springer, Heidelberg (2010)
7. INRIA: Yael library: Optimized implementations of computationally demanding functions (2009), https://gforge.inria.fr/projects/yael/
8. Jiang, Y.G., Bhattacharya, S., Chang, S.F., Shah, M.: High-level event recognition in unconstrained videos. IJMIR, 1–29 (2012)
9. Joachims, T.: Optimizing search engines using clickthrough data. In: SIGKDD, pp. 133–142. ACM (2002)
10. Li, W., Yu, Q., Divakaran, A., Vasconcelos, N.: Dynamic pooling for complex event recognition. In: ICCV (2013)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
12. Natarajan, P., Wu, S., Vitaladevuni, S., Zhuang, X., Tsakalidis, S., Park, U., Prasad, R.: Multimodal feature fusion for robust event detection in web videos. In: CVPR (2012)
13. Niebles, J.C., Chen, C.-W., Fei-Fei, L.: Modeling temporal structure of decomposable motion segments for activity classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 392–405. Springer, Heidelberg (2010)
14. Oneata, D., Verbeek, J., Schmid, C.: Action and event recognition with fisher vectors on a compact feature set. In: ICCV, pp. 1817–1824 (2013)
15. Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Kraaij, W., Smeaton, A.F., Quenot, G.: Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In: Proceedings of TRECVID 2013. NIST (2013)
16. Quattoni, A., Carreras, X., Collins, M., Darrell, T.: An efficient projection for $l_{1,\infty}$, infinity regularization. In: ICML (2009)
17. Rakotomamonjy, A.: Sparse support vector infinite push. In: ICML (2012)
18. Rudin, C.: The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. JMLR 10, 2233–2271 (2009)
19. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. CRCV-TR-12-01 (2012)
20. Tamrakar, A., Ali, S., Yu, Q., Liu, J., Javed, O., Divakaran, A., Cheng, H., Sawhney, H.: Evaluation of low-level features and their combinations for complex event detection in open source videos. In: CVPR (2012)
21. Tang, K., Fei-Fei, L., Koller, D.: Learning latent temporal structure for complex event detection. In: CVPR (2012)
22. Vahdat, A., Cannons, K., Mori, G., Oh, S., Kim, I.: Compositional models for video event detection: A multiple kernel learning latent variable approach. In: ICCV, pp. 1185–1192 (2013)
23. Vedaldi, A., Fulkerson, B.: Vlfeat: An open and portable library of computer vision algorithms (2008), http://www.vlfeat.org/
24. Wang, H., Klaser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR (2011)
25. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. IJCV, 1–20 (2013)