2014 IEEE Conference on Computer Vision and Pattern Recognition

Video Event Detection by Inferring Temporal Instance Labels

Kuan-Ting Lai[§†], Felix X. Yu[‡], Ming-Syan Chen[§†] and Shih-Fu Chang[‡]
[§]Graduate Institute of Electrical Engineering, National Taiwan University, Taiwan
[†]Research Center for IT Innovation, Academia Sinica, Taiwan
[‡]Department of Electrical Engineering, Columbia University, USA
{ktlai, mschen}@arbor.ee.ntu.edu.tw, {yuxinnan, sfchang}@ee.columbia.edu

Abstract

Video event detection allows intelligent indexing of video content based on events. Traditional approaches extract features from video frames or shots, then quantize and pool the features to form a single vector representation for the entire video. Though simple and efficient, the final pooling step may lead to loss of temporally local information, which is important in indicating which part in a long video signifies presence of the event. In this work, we propose a novel instance-based video event detection approach. We represent each video as multiple "instances", defined as video segments of different temporal intervals. The objective is to learn an instance-level event detection model based on only video-level labels. To solve this problem, we propose a large-margin formulation which treats the instance labels as hidden latent variables, and simultaneously infers the instance labels as well as the instance-level classification model. Our framework infers optimal solutions that assume positive videos have a large number of positive instances while negative videos have the fewest ones. Extensive experiments on large-scale video event datasets demonstrate significant performance gains. The proposed method is also useful in explaining the detection results by localizing the temporal segments in a video which is responsible for the positive detection.

1. Introduction

Video event detection is useful in many applications such as video search, consumer video analysis, personalized advertising, and video surveillance, to name a few [15]. Many methods has been proposed for detecting video events, including large margin based method, graphical model, knowledge-based techniques, *etc*. [9]. The most commonly used approach is to represent a video as a global Bag-of-Word (BoW) vector [18]. The BoW method can be divided into three stages: First, the local features (visual, audio or attributes) are extracted from frames or segments of a video. The features are then quantized based on a learned codebook, or dictionary [24]. Finally, the quantized features are pooled on the whole video to form a global vector representation.

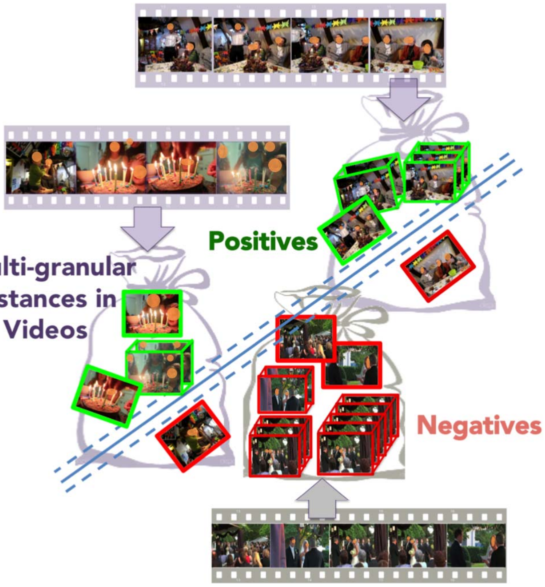


Figure 1. Illustration of the proposed framework. The event "birthday party" can be recognized by instances containing "birthday cake" and "blowing candles". Our method simultaneously infers hidden instance labels and instance-level classification model (the separating hyperplane) based on only video-level labels.

Representing a video as a single vector is simple and efficient. Unfortunately, much information may be lost in the final pooling step, leading to unsatisfactory performance. In fact, a video is comprised of multiple "instance", such as frames and shots. Some instances contain key evidences of the event being considered. For example, event like "birthday party" may be well detected by frames containing cakes, and candles, and "parkour" may be well detected by shots of person jumping up and down on the street. As

1063-6919/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPR.2014.288

shown in recent research, human can well exploit key evidences and recognize events by only short video clips [2]. Pooling the features of the video into a single aggregate form may fail to take advantage of such rich cues over multiple instances. Intuitively, by considering the instances of the videos, more distinctive event patterns can be learned, and therefore better event recognition can be achieved.

Motivated by the above facts, we study instance-based video classification, as illustrated in Fig. 1. Each video contains multiple "instances", defined as video segments of different temporal lengths. The definitions of the instances may be flexible - they can be video frames, fixed-length videos, video shots detected based on content changes, and the whole video. In this paper, we propose to consider instances of multiple granularities, *i.e.* diverse instances of various lengths in videos. This gives us the flexibility in modeling video events of different temporal scales: some may be of short durations while others may involve long intervals. Figure 2 shows the multiple granularities of the instances in a video. Our goal is to learn an instance-level event detection model, while assuming instance labels are not available due to the prohibitive cost in annotating the instances. The conventional method based on Multiple-Instance Learning (MIL) may appear to be a natural choice. But due to its over simplified model in transferring instance prediction to bags, it is not a satisfactory solution, as will be shown and evaluated later in the paper.

To solve this challenging problem, we propose a large-margin framework. It treats the instance labels as latent variables, and simultaneously infers the hidden instance-labels as well as the instance-level classification model. Our key assumption is that the positive videos usually have a large portion of positive instances, while the negative videos have few positive instances. The proposed method not only leads to more accurate event detection results, but also learns the instance-level detector, explaining when and why certain event happens in the video.

Our paper includes the following major contributions:

- We propose a novel instance-based video event detection method (Section 3).
- Based on a large-margin learning framework, we develop an algorithm which can simultaneously infer the instance labels and the instance-level event detection model from only video-level labels (Section 3 and 4).
- Extensive experiment evaluations demonstrate the superior performance of our method on large-scale video datasets (Section 6).

2. Related Works

2.1. Video Event Detection

Video event detection is a widely studied topic in computer vision. A good survey of state-of-the-arts was made in [9]. Generally speaking, video event detection system can be divided into three stages: feature extraction, feature quantization and pooling, training/recognition.

One focus of previous research is on designing new features, including low-level features of visual features [6, 12], action features [22], audio features [14], and mid-level representation including concept feature, attributes [20] *etc*. There are also significant efforts on improving the event recognition modeling, such as max-margin based methods, graphical models and some knowledge based techniques, as reviewed in [9]. However, most former approaches rely on a global vector to represent one video. The global approach neglects important local information of the events. Recently some researchers attempted to address this problem and proposed several new algorithms. Tang *et al*. [19] treat video segments as latent variables and adopted variable-duration hidden Markov model to represent events. Cao *et al*. [3] proposed scene aligned pooling, which divides videos into shots with different scenes, and pooling local features under each scene. Li *et al*. [11] proposed dynamic pooling, which employs various strategies to split videos into segments based on temporal structures. Different from their methods, which focus on exploiting temporal structures for pooling, our framework focuses on learning "instance" labels. The proposed approach can also be seen as complementary to the above pooling strategies, for which the video instances can be formed by dynamic pooling or scene aligned pooling.

2.2. Multiple-Instance Learning

In order to use local patterns in a video, one readily available learning method is Multiple-Instance Learning (MIL) [7]. In MIL, the training data is provided in "bags". And the labels are only provided on the bag-level. A bag is labeled as positive *iff.* one or more instances inside the bag are positive. In computer vision, MIL has been applied in scene classification [13], content-based image retrieval [27], and image classification [5]. The two most popular algorithms for MIL are mi-SVM and MI-SVM [1]. The first algorithm emphasizes searching max-margin hyperplanes to separate positive and negative instances, while the second algorithm selects the most representative positive instance for each positive bag during optimization iterations, and concentrates on bag classification.

In event detection, a video can be seen as a bag containing multiple instances. The labels are only provided on video-level. Therefore, algorithms of MIL can be directly applied. However, existing algorithms of MIL are not suitable for video event classification. One restriction is that MIL relies on a single instance (often computed based on the max function) in prediction, making the method very sensitive to false alarm outliers; another drawback is that it assumes that negative bags have no positive instances, lead-

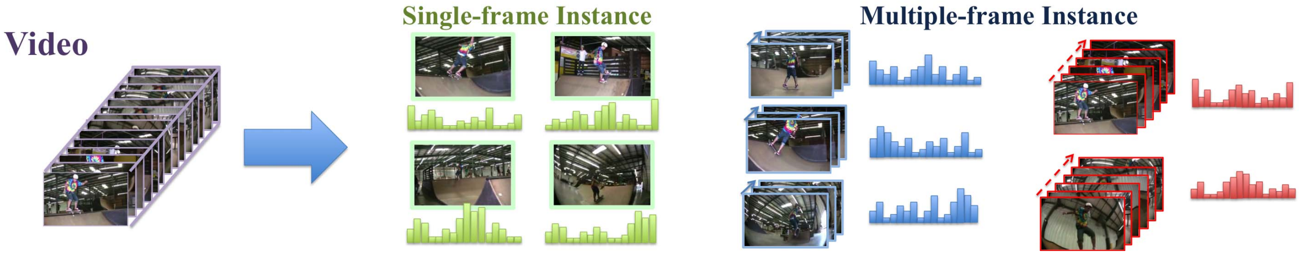


Figure 2. Illustration of multiple-granular instances with different time lengths. In our framework, the instance of minimum granularity is a frame, and instance of maximum granularity is the whole video. Each instance is represented by pooling the included frame-level BoWs.

ing to unstable results for complicated events.

2.3. Learning with Label Proportions

Several methods have been proposed to address the limitations of MIL. Chen *et al*. [4] proposed to embed bags into instance space via instance similarity measure; Zhang *et al*. [26] proposed a new method which considers both local (instance) and global (bag) feature vectors. Another generalization of MIL is Learning with Label Proportions (LLP). In LLP, the learner has access to the proportion of positive instances of each bag. Compared to MIL, LLP can produce more stable results, as the model does not rely on just a single instance per bag. Several methods have been studied for LLP [16, 25].

Our video recognition algorithm is inspired by proportion SVM (p-SVM or $\propto$SVM) [25], which explicitly models the latent unknown instance labels together with the known label proportions in a large-margin framework. The $\propto$SVM was shown to outperform other alternatives. Different from $\propto$SVM, in video classification, the exact label proportion is unknown. Our key assumption is that a large portion of instances in a positive video should be positive, whereas few instances in the negative videos may be positive. We also consider multiple time granularities to form the instances, leading to significant performance improvement in video classification.

3. The Proposed Method

Setting. Suppose we have the training dataset $\{V_m\}_{m=1}^M$. Considering a single event, in each video V_m, there are N_m instances $\{\mathbf{x}_i^m, y_i^m\}_{i=1}^{N_m}$, in which $\mathbf{x}_i^m$ is the feature vector of the i-th instance in the m-th video, and $y_i^m \in \{-1, 1\}$ is corresponding event label. Here, $y_i^m = 1$ if the instance is positive for the event, and $y_i^m = -1$ otherwise. As pointed out in Section 1, for most of the cases, the instance labels are unknown, and the supervised information is only provided on video-level. Therefore, we propose to learn an instance-level classification model based on only the video-level supervised information.

In Section 3.1 , we propose to use a formulation similar to $\propto$SVM for a simple case in which the proportion of positive instances for each video is known. In Section 3.2, we show how to extend the method to the real-world case, in which only the binary video-level event label is given. In Section 3.3, we propose to use multi-granular instances for improved video classification.

3.1. Event Recognition by Instance Proportions

In this section, we consider a simple case that the *proportion* of positive instances $\{P_m\}_{m=1}^M$ for each video is known. Here, $P_m \in [0, 1]$ is the positive instance proportion for the m-th video V_m. The target is to train an instance-level classifier to classify individual instances. We propose to learn a large margin event classification model $(\mathbf{w}, b)$, such that an instance $\mathbf{x}$ is predicted to be positive if $(\mathbf{w}^\top \mathbf{x} + b) > 0$, and negative if $(\mathbf{w}^\top \mathbf{x} + b) \leq 0$. In order to solve the problem, we propose to jointly inferring the instance labels and the prediction model. Our formulation is in parallel with the $\propto$SVM [25], which tries to find a large-margin classifier, compatible with the given label proportions. We note that, given the instance labels $\mathbf{y}^m$, the positive instance proportion $p_m(\mathbf{y}^m)$ of the m-th video can be expressed as:

$$p_m(\mathbf{y}^m) = \frac{\sum_{i=1}^{N_m} I_{(y_i^m = 1)}}{N_m}, \tag{1}$$

where $\mathbf{y}^m = [y_1^m, \cdots, y_{N_m}^m]$. $I_{(\cdots)}$ is the indicator function which is 1 when the argument is true or 0 otherwise. The above is equivalent to the following:

$$p_m(\mathbf{y}^m) = \frac{\sum_{i=1}^{N_m} y_i^m}{2N_m} + \frac{1}{2}, \tag{2}$$

The parameters of the classification model and unknown instance labels are jointly learned by optimizing the following objective function:

$$\min_{\{\mathbf{y}^m\}_{m=1}^M, \mathbf{w}, b} \frac{1}{2}||\mathbf{w}||^2 + C \sum_{m=1}^{M} \sum_{i=1}^{N_m} L\big(y_i^m, (\mathbf{w}^\top \mathbf{x}_i^m + b)\big)$$
$$s.t. \quad p_m(\mathbf{y}^m) = P_m, \quad m = 1, \cdots, M. \tag{3}$$

The first term is the classic SVM term to find a max margin separating hyperplanes of the two classes. The second term $L(\cdot)$ is the empirical loss function of instance labels and predictions. The proposed framework permits choosing different loss functions for $L(\cdot)$. Throughout this paper, the hinge loss function is used for $L(\cdot)$, where

$$L\big(y_i^m, \mathbf{w}^\top \mathbf{x}_i^m + b\big) = \max(0, 1 - y_i^m(\mathbf{w}^\top \mathbf{x}_i^m + b)). \quad (4)$$

In summary, the framework tries to find a large margin classifier, compatible with the given label proportions. As a special case, if we known all the instance labels $y_i^m, m = 1, \cdots, M, i = i, \cdots, N_m$, the framework becomes classic supervised SVM.

3.2. Dealing with the Unknown Proportion

In the previous section, we discussed the case when the instance label proportion $\{P_m\}_{m=1}^M$ are known. However, in video event classification, we only know the video-level binary labels $\{Y_m\}_{m=1}^M$, in which $Y_m \in \{-1, 1\}$ is the video-level label of V_m.

To solve this problem, our key assumption is that *each positive video contains "many" positive instances, while each negative video contains few or no positive instances.* Specifically, we propose the following modified formulation to achieve the above assumption:

$$\min_{\{\mathbf{y}^m\}_{m=1}^M, \mathbf{w}, b} \quad \frac{1}{2}||\mathbf{w}||^2 + C \sum_{m=1}^{M} \sum_{i=1}^{N_m} L\big(y_i^m, (\mathbf{w}^\top \mathbf{x}_i^m + b)\big)$$
$$+ \quad C_p \sum_{m=1}^{M} |p_m(\mathbf{y}^m) - P_m| \qquad (5)$$
$$s.t. \quad P_m = \begin{cases} 1 & \text{if } Y_m = 1 \\ 0 & \text{if } Y_m = -1 \end{cases}, m = 1, \cdots, M.$$

The first modification is to move hard constraint of Eq. 3 to the objective function: the third term is a loss function to penalize the difference between target positive instance proportion P_m and estimated proportion $p_m(\mathbf{y}^m)$. Secondly, we set the positive instance proportion of positive videos to 1 and negative videos to 0. Under this setup, the framework encourages large proportions of positive instances in positive videos while penalizes the positive instances in negative videos. C_p is the parameter to control the "strength" of our assumption. In practice, C_p can be tuned based on cross validation.

3.3. Instances with Multiple Granularities

One key question left unanswered is how to design the instances for each video. The instances can be frames, shots, video segments or even whole videos. Instances with different temporal lengths can be useful for recognizing different events. For example, "birthday party" can be identified by single frames containing cakes and candles, whereas sport-like actions such as "attempting board trick" and "parkour" are better detected by video segments characterizing actions.

Motivated by the observation, we consider instances of multiple granularities based on different length of time intervals. The feature representation of multiple-granular instances are obtained by pooling the local features into segment-level BoW with specific time lengths. Note that the video BoW is one special case in our framework.

The original $\propto$SVM framework treats all instances equally and can not differentiate instances of multiple granularities. Therefore, we develop a new formula which can assign weights to different granular instances. Our proposed formula is introduced below.

Suppose we have K granularities. The number of total instances for the k-th granularity of the m-th video is N_k^m. We define a label vector $\mathbf{y}_k^m = [(y_1)_k^m, \cdots, (y_{N_k^m})_k^m]$. The component $(y_i)_k^m$ is the i-th instance label of k-th granularity in m-th video. The weight for the k-th granularity is defined as t_k. Therefore, we write the new proportion function $p_m(\mathbf{y}_1^m \cdots \mathbf{y}_K^m)$ as:

$$p_m(\mathbf{y}_1^m \cdots \mathbf{y}_K^m) = \frac{\sum_{k=1}^K t_k(\mathbf{1}^\top \mathbf{y}_k^m)}{2\sum_{k=1}^K \ t_k N_k^m} + \frac{1}{2}. \qquad (6)$$

The total number of instances in m-th video N_m is now the weighted sum of instances at all granularities $\sum_{k=1}^K t_k N_k^m$. In this paper, the feature of instances from the k-th granularity is computed by averaging the BoW representation of the included frames, and the weights are simply set as the number of frames included. We define the i-th feature vector of k-th granularity and m-th video as $(\mathbf{x}_i)_k^m$. The weighted version of Eq. (5) becomes

$$\min_{\{\mathbf{y}^m\}_{m=1}^M, \mathbf{w}, b} \quad \frac{1}{2}||\mathbf{w}||^2 + C_p \sum_{m=1}^{M} |p_m(\mathbf{y}_1^m \cdots \mathbf{y}_K^m) - P_m|$$
$$+ \quad C \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{i=1}^{N_k^m} t_k L\big((y_i)_k^m, (\mathbf{w}^\top (\mathbf{x}_i)_k^m + b)\big)$$
$$s.t. \quad P_m = \begin{cases} 1 & \text{if } Y_m = 1 \\ 0 & \text{if } Y_m = -1 \end{cases}, m = 1, \cdots, M. \qquad (7)$$

Solving the above equation is a challenging problem, since it is NP-hard combinatorial optimization problem that cannot be solved in polynomial time. In next section, we will explain our strategy and elaborate each step of the optimization process.

4. Optimization Procedure

In order to "solve" Eq. (7), we apply the alternating optimization to to find a local suboptimal solution:

- First we fix instance labels $\{\mathbf{y}^m\}_{m=1}^M$ and solve $\mathbf{w}$ and b. By fixing $\{\mathbf{y}^m\}_{m=1}^M$, the optimization problem becomes a classic weighted SVM

$$\min_{\mathbf{w}, b} \frac{1}{2}||\mathbf{w}||^2 + C \sum_{m=1}^{M} \sum_{k=1}^{K} \sum_{i=1}^{N_k^m} t_k L\big((y_i)_k^m, (\mathbf{w}^\top (\mathbf{x}_i)_k^m + b)\big).$$

- Second we fix $\mathbf{w}$ and b and update instance labels $\{\mathbf{y}^m\}_{m=1}^M$. The problem now becomes:

$$\min_{\{\mathbf{y}_m\}_{m=1}^M} \quad C\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i=1}^{N_k^m} t_k L\big((y_i)_k^m, (\mathbf{w}^\top(\mathbf{x}_i)_k^m + b)\big) + \ C_p \sum_{m=1}^{M} |p_m(\mathbf{y}_1^m \cdots \mathbf{y}_K^m) - P_m|. \qquad (8)$$

Due to the fact that each video V_m contributes to the objective independently, we can optimize (8) one video at a time. The procedures of optimizing video V_m are as follows. First, we set all instance labels $(y_i)_k^m$ in video V_m to -1, and calculate each empirical loss increase $(\delta_i)_k^m$ by flipping each $(y_i)_k^m$ to 1. The value of $(\delta_i)_k^m$ can be computed as

$$(\delta_i)_k^m = \big(1-(\mathbf{w}^\top g_k(\mathbf{x}_i^m)+b)\big)_+ - \big(1+(\mathbf{w}^\top g_k(\mathbf{x}_i^m)+b)\big)_+,$$

where function $(x)_+ = \max(x, 0)$.

Once all empirical loss increase $(\delta_i)_k^m$ are computed, all weighted loss value $t_k(\delta_i)_k^m$ are then sorted by descending order. This part is the same as original $\propto$SVM. However, searching minimum total loss is a different problem in our formula. In $\propto$SVM the instance labels are flipped one by one to calculate the proportion loss increases, and the number of labels with minimum total loss are selected to be flipped. When instances have different weights, there are more than one combination that can achieve certain proportion. In our framework, we employ a greedy algorithm that can search a sub-optimal solution in log-linear time.

The proposed optimization procedure is shown in Algorithm 1. The objective function is non-increasing in our optimization process. The algorithm stops when the reduction of objective function is less than certain threshold, which is set to 10^{-2} in our experiments. Empirically, the optimization process converges fast within just tens of iterations. Although the above method is based on linear large-margin framework, it can be easily extended to kernel scenario by applying kernel trick when solving $(\mathbf{w}, b)$ with fixed instance labels.

5. Discussions

5.1. Event Detection at Video Level

In the previous section, we propose to learn an event detection model on the instance level, based on video-level labels. One intrinsic advantage of our method is that it can naturally discover the key evidences which support the existence of specific events. The top ranked 16 evidences selected by our method are shown in Fig. 4 and Fig. 8. Some selected single-frame instances are strong evidences, by which human can confirm the existence of target event by seeing those frames. In order to perform event detection on video level, we can first apply the instance classifier on all instances of the test videos. The video-level detection score can then be obtained by performing weighted average of all instance scores. Intuitively, a video containing more positive instances tend to have higher probability of being positive. We will later show by experiment that our approach can lead to significant performance improvement for video-level event detection.

Algorithm 1 Optimization Procedure

```
Input: $k = 1 \cdots K, m = 1 \cdots M$
  video label $Y_m \in \{1, -1\}$.
  instance $\mathbf{x}_k^m$, instance weight $t_k \in \mathbb{R}$.
  proportion $P_m = 1$ if $Y_m = 1$, $P_m = 0$ if $Y_m = -1$.
  convergence threshold $\theta = 0.01$.
Initialization:
  $(y_i)_k^m \leftarrow Y_m, i = 1 \cdots N_k^m, k = 1 \cdots K, m = 1 \cdots M$.
repeat
  fix $\mathbf{y}_k^m$ and solve $\mathbf{w}$ and b.
  set cost reduction $C_R \leftarrow 0$.
  for $m = 1 \cdots M$ do
    $(y_i)_k^m \leftarrow -1, k = 1 \cdots K, i = 1 \cdots N_k^m$
    compute all $(\delta_i)_k^m$ for $(y_i)_k^m$ (Eq. 9)
    sort $(y_i)_k^m$ by $t_k(\delta_i)_k^m$ in descending order
    for sorted $(y_i)_k^m$ do
      flip $(y_i)_k^m$, calculate the loss reduction incrementally.
    end for
    select maximum loss reduction.
    flip the labels to get max loss reduction, and update $C_R$.
  end for
until convergence ($C_R < \theta$)
Output: $\mathbf{w}, b, \mathbf{y}$
```

5.2. Computational Cost

Because we are using a $\propto$SVM-like algorithm, the computation complexity (with linear SVM solver) is $\mathcal{O}(N \log \max_m(N_m))$, in which $\max_m(N_m)$ is the maximum number of instances in m-th video, and N is the number of total instances. The formula can be written as $\mathcal{O}(\mathcal{V}\mathcal{T} \log \max_m(N_m))$, in which $\mathcal{V}$ is the total number of videos and $\mathcal{T}$ is the averaged number of instances per video. As $\mathcal{T} \log \max_m(N_m)$ can be seen as a constant, the computational complexity is the same as video-based event classification with linear SVM.

In practice, several techniques can be applied to improve the computational time. For example, the framework can be improved further by solving the SVM in their inner loops incrementally. One approach is to utilize warm start and partial active-set methods proposed by Shilton *et al.* [17]. Another method is to employ non-linear kernels using explicit feature maps [21], so that the complexity of our method can become linear even with certain nonlinear kernels.

5.3. Learning with Heterogeneous Instances

In the previous section, we are considering the multiple granularities of instances with the same underlying feature representation. In practice, the instances may come with different representations. For example, we may have instances represented by image/audio/action features respectively. In such case, the proposed approach can be applied with minor changes to learn a classification model for each type of feature representation. We can also jointly learn the classification models with a modified objective function. We leave this task to our future work.

6. Experiments

Datasets. To evaluate our framework, we conducted experiments on three large-scale video datasets: TRECVID Multimedia Event Detection (MED) 2011, MED 2012 [15] and Columbia Consumer Videos (CCV) [10] datasets. All our experiments are based on SIFT feature and linear SVM.

Features. In this paper, we selected SIFT [12] as underlying local features for initial evaluation. Note that our method can be easily extended to include multiple features by using fusion techniques. For example, we can train different instance-based detection models for each feature independently, and fuse detected scores of detectors using different features for final event detection. Additionally, by employing multiple features, we can discover unique cues, *e.g.* actions, colors, audio, for each video event.

Settings. For each video, we extract frames at every 2 seconds. Each frame is resized to 320×240, and SIFT features are extracted densely with 10-pixel step by VLFeat library. The frame features are then quantized into 5000 Bag-of-Word vector. The frame-level SIFT BoW is taken as instance feature vector. The liblinear SVM tool [8] is applied to solve $\mathbf{w}$ and b while instance labels $\mathbf{y}$ are fixed. The cost parameters C and C_p are chosen from the scale of $\{0.01, 0.1, 1, 10, 100\}$ based on cross-validation.

Baselines. We evaluate four baseline algorithms on the dataset: mi-SVM, MI-SVM [1], video BOW, and proportional SVM ($\propto$SVM or p-SVM) [25] with single frame instance. The mi-SVM and MI-SVM both utilize Multiple Instance Learning. As introduced in Section 2.2, the mi-SVM focuses on instance-level while MI-SVM focuses on bag-level classification. In practice, the mi-SVM infers the instance labels iteratively while forcing all instances in negative videos to be negative. The MI-SVM selects one positive instance with the highest score in each positive video during each iteration, and forces all instances in negative videos to negative. We adopt the MILL library [23] and modified it for better performance. For $\propto$SVM, we set unknown proportions as in Section 3.2 and chose only single frames as instances [1].

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	Sewing project		

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

6.1. Columbia Consumer Videos (CCV)

The Columbia Consumer Video (CCV) benchmark defines 20 events and contains 9,317 videos downloaded from YouTube. The event names and train/test splits can be found in the original paper [10].

In terms of selecting multi-granular instances, we evaluated combinations of various instances on CCV dataset. Empirically, we found that more granularities lead to better performance. For example, the result of using four granularities (single frame, 3-frame shot, and 5-frame shot and whole video) achieves best results with mAP 0.436. However, increasing the number of granularities will cause higher computation cost. Considering a trade-off between time and performance, in this paper we only use two granularities: single-frame and whole video instances, in all the experiments.

The experimental results are shown in Fig. 3. The mi-SVM and MI-SVM are inferior to standard video-level BoW method. It is due to the restrictive assumption of MIL which focuses on searching one most representative instance in each video and treats all instances in negative video as negatives. On the contrary, the $\propto$SVM doesn't make this assumption and outperforms video BoW. Our method further improves the performance by considering multi-granular instances and relatively outperforms linear video BoW and $\propto$SVM by 10.2% and 4.8%.

6.2. TRECVID MED12

The MED12 dataset contains 25 complex events and 5,816 videos. The names for both MED 11 and MED 12 are listed in Table 1. We split two-third of the data as training set (3,878 videos) and use the rest as test set (1,938 videos). The average number of extracted frames in each video is 79.4, and the average learning time of one event on a single Intel Xeon CPU @2.53GHz is around 40 minutes. The experimental results are shown in Fig. 5. Comparisons of the results are found similar to those observed for the CCV dataset. The mi-SVM and MI-SVM are inferior

[1] `https://github.com/felixyu/pSVM`

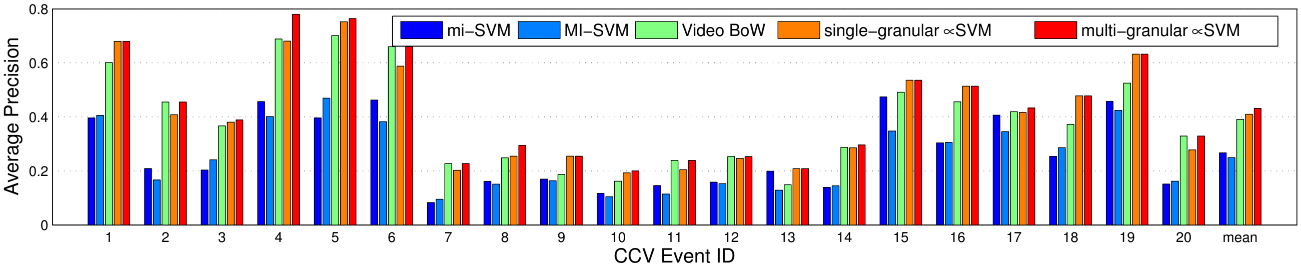
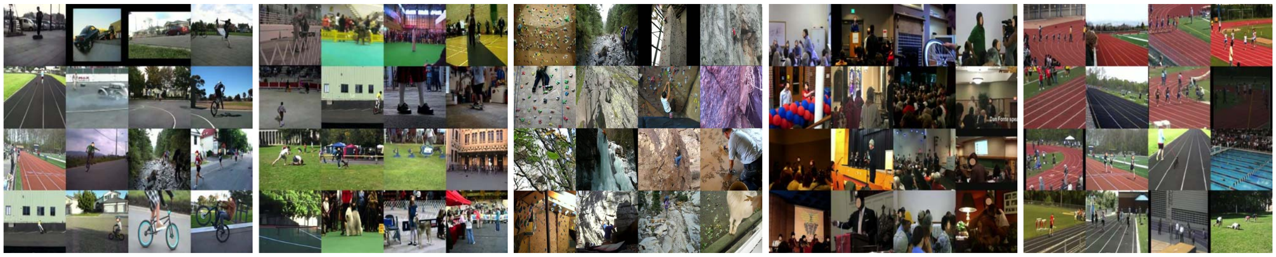


Figure 3. Experimental results of 20 complex events in Columbia Consumer Videos (CCV) dataset. The mean APs are 0.26 (mi-SVM), 0.25 (MI-SVM), 0.39 (Video BoW), 0.41 (single-granular $\propto$SVM) and 0.43 (multi-granular $\propto$SVM).



(a) attempting bike trick (b) dog show (c) rock climbing (d) town hall meeting (e) win race without vehicles

Figure 4. The top 16 key positive frames selected for the events in MED12. The proposed method can successfully detect important visual cues for each event. For example, the top ranked instances of "winning race without vehicles" are about tracks and fields.

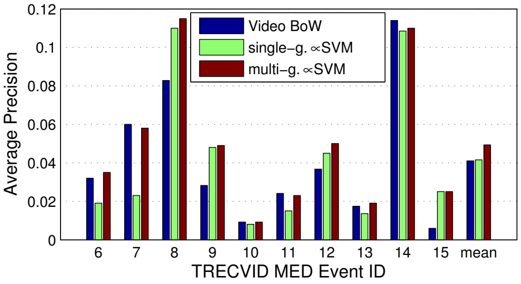


Figure 6. The APs from Event 6 to Event 15 in MED 2011.

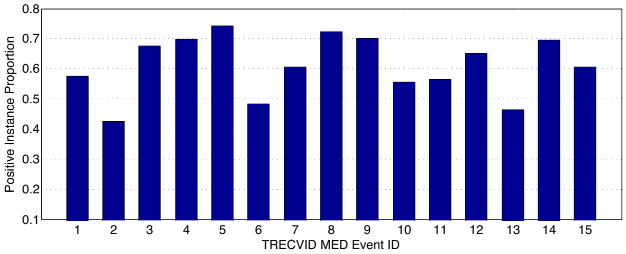


Figure 7. The proportions of positive instances learned by our method in positive training videos of MED11 events

to the standard video-level BoW method. Our method relatively outperform video BoW and $\propto$SVM by 21.4% and 9.68%, respectively. As mentioned earlier, our method also offers great benefits in pinpointing the specific local segments that signify the events. Figure 4 shows the automatically selected key frames in videos that is detected as positive, which can be used to explain the detection result.

6.3. TRECVID MED11

In this experiment, we follow the official data splits of TRECVID MED contest. NIST provided three data splits of MED11: event collection (EC), the development collection (DEVT) and test collection (DEVO). The event collection contains 2,680 training videos over 15 events. The DEVT set with 10,403 videos was released for contestants to evaluate their systems. The final performance are evaluated on DEVO set with 32,061 test videos. The average number of extracted frames per video is 59.8, and the average learning time of one event on a single Intel Xeon CPU @2.53GHz is around 6 hours. The experimental results are shown in Fig. 6. Because the DEVO set does not include any video of Event 1 to Event 5, only results of Event 6 to Event 15 are reported. The $\propto$SVM outperforms Video BOW on "Flash mob gathering", "Getting vehicle unstuck", and "Parade", but produced worse results for other events. This is an interesting finding as it confirms that instances of different lengths are needed for representing different events. Our method outperforms other methods by around 20% in this experiment. Figure 7 illustrates the proportions of positive instances learned for each event. The optimal positive instance proportion can be as low as 42.6% despite the 100% target set in Eq. (7). Some top-ranked frame instances learnt by our method are shown in Fig. 8.

7. Conclusion

We propose a novel approach to conduct video event detection by simultaneously inferring instance labels, and learning the instance-level event detection model. The

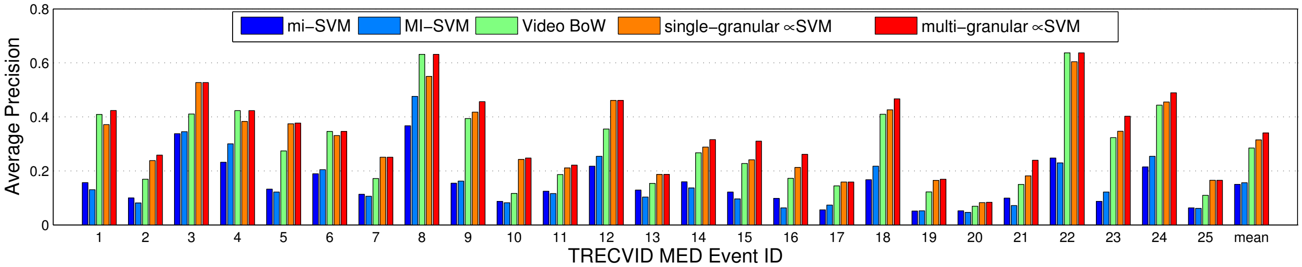
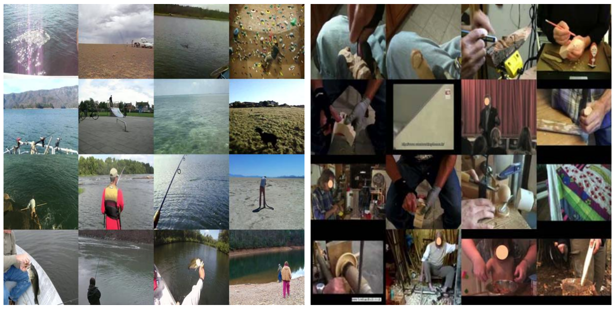


Figure 5. Evaluation results of 25 complex events in TRECVID MED 12 video dataset. The mean APs are 0.15 (mi-SVM), 0.16 (MI-SVM), 0.28 (Video BoW), 0.31 (∝SVM) and 0.34 (Our method).



(a) landing a fish (b) woodworking project

Figure 8. The top 16 key positive frames selected by our algorithm for some events in TRECVID MED11.

proposed method considers multiple granularities of instances, leveraging both local and global patterns to achieve best results, as clearly demonstrated in extensive experiments. The proposed methods also provide intuitive explanation of detection results by localizing the specific temporal frames/segments that signify the presence of the event.

References

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

[2] S. Bhattacharya, F. X. Yu, and S.-F. Chang. Minimally needed evidence for complex event recognition in unconstrained videos. In *ICMR*, 2014.

[3] L. Cao, Y. Mu, A. Natsev, S.-F. Chang, G. Hua, and J. R. Smith. Scene aligned pooling for complex video recognition. In *ECCV*. 2012.

[4] Y. Chen, J. Bi, and J. Z. Wang. Miles: Multiple-instance learning via embedded instance selection. *PAMI*, 28(12):1931–1947, 2006.

[5] Y. Chen and J. Z. Wang. Image categorization by learning and reasoning with regions. *JMLR*, 5:913–939, 2004.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[7] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31–71, 1997.

[8] 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.

[9] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah. High-level event recognition in unconstrained videos. *IJMIR*, pages 1–29, 2012.

[10] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *ICMR*, 2011.

[11] W. Li, Q. Yu, A. Divakaran, and N. Vasconcelos. Dynamic pooling for complex event recognition. In *ICCV*, 2013.

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[13] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *ICML*, 1998.

[14] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. *PAMI*, 116:374–388, 1976.

[15] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2013*. NIST, 2013.

[16] S. Rüeping. SVM classifier estimation from group probabilities. In *ICML*, 2010.

[17] A. Shilton, M. Palaniswami, D. Ralph, and A. C. Tsoi. Incremental training of support vector machines. *IEEE Transactions on Neural Networks*, 16(1):114–131, 2005.

[18] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[19] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012.

[20] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*. 2010.

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

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

[23] J. Yang. Mill: A multiple instance learning library, 2009.

[24] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[25] F. X. Yu, D. Liu, S. K., T. Jebara, and S.-F. Chang. ∝SVM for learning with label proportions. In *ICML*, 2013.

[26] D. Zhang, J. He, L. Si, and R. D. Lawrence. Mileage: Multiple instance learning with global embedding. In *ICML*, 2013.

[27] Q. Zhang, S. A. Goldman, W. Yu, and J. E. Fritts. Content-based image retrieval using multiple-instance learning. In *ICML*, 2002.