A Generic Framework for Event Detection in Various Video Domains

Tianzhu Zhang[1,2], Changsheng Xu[1,2], Guangyu Zhu[3], Si Liu[1,2,3], Hanqing Lu[1,2]
[1] National Lab of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100190, P. R. China
[2] China-Singapore Institute of Digital Media, Singapore 119615, Singapore
[3] Department of Electrical and Computer Engineering, National University of Singapore, Singapore
{tzzhang, csxu, sliu, luhq}@nlpr.ia.ac.cn, elezhug@nus.edu.sg

ABSTRACT

Event detection is essential for the extensively studied video analysis and understanding area. Although various approaches have been proposed for event detection, there is a lack of a generic event detection framework that can be applied to various video domains (e.g. sports, news, movies, surveillance). In this paper, we present a generic event detection approach based on semi-supervised learning and Internet vision. Concretely, a ***Graph-based Semi-Supervised Multiple Instance Learning (GSSMIL)*** algorithm is proposed to jointly explore small-scale expert labeled videos and large-scale unlabeled videos to train the event models to detect video event boundaries. The expert labeled videos are obtained from the analysis and alignment of well-structured video related text (e.g. movie scripts, web-casting text, close caption). The unlabeled data are obtained by querying related events from the video search engine (e.g. YouTube) in order to give more distributive information for event modeling. A critical issue of GSSMIL in constructing a graph is the weight assignment, where the weight of an edge specifies the similarity between two data points. To tackle this problem, we propose a novel ***Multiple Instance Learning Induced Similarity (MILIS)*** measure by learning instance sensitive classifiers. We perform the thorough experiments in three popular video domains: movies, sports and news. The results compared with the state-of-the-arts are promising and demonstrate our proposed approach is performance-effective.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing - Abstracting methods, Indexing methods.

General Terms

Algorithm, Measurement, Performance, Experimentation

Keywords

Event Detection, Graph, Multiple Instance Learning, Semi-supervised Learning, Broadcast Video, Internet, Web-casting Text

MM'10, October 25–29, 2010, Firenze, Italy.

1. INTRODUCTION

With the explosive growth of multimedia content on broadcast and Internet, it is urgently required to make the unstructured multimedia data accessible and searchable with great ease and flexibility. Event detection is particularly crucial to understanding video semantic concepts for video summarization, indexing and retrieval purposes. Therefore, extensive research efforts have been devoted to event detection for video analysis [23, 26, 33].

Most of the existing event detection approaches rely on video features and domain knowledge, and employ labeled samples to train event models. The semantic gap between low-level features and high-level events of different kinds of videos, the ambiguous video cues, background clutter and variant changes of camera motion, etc., further complicate the video analysis and impede the implementation of event detection systems. Moreover, due to the diverse domain knowledge in different video genres and insufficient training data, it is difficult to build a generic framework to unify event detection in different video domains (e.g. sports, news, movies, surveillance) with a high accuracy.

To solve these issues, most of techniques for event detection currently rely on video content and supervised learning in the form of labeled video clips for particular classes of events. It is necessary to label a large amount of samples in the training process to achieve good detection performance. In order to reduce the human labor-intension, one can exploit the expert supervisory information in text source [3, 15, 23], such as movie scripts, web-casting text and closed captions, which can provide useful information to locate possible events in video sequences. However, it is very cost-expensive and time-consuming to collect large-scale training data by text analysis, and there are still many videos without the corresponding text information for use. The Internet, nevertheless, is a rich information source with many event videos taken under various conditions and roughly annotated. For example, the surrounding text is an important clue used by search engines. Our intuition is that it is convenient to obtain a large-scale collection of videos as unlabeled data to improve the performance of event detection. By doing this, we propose a semi-supervised learning framework to exploit the expert labeled and unlabeled video data together.

However, the labeled data by text analysis only have the weakly associated labels, which means we know the video's label, but there may be no precise information about the localization of event in video. For the unlabeled data by Internet searching, we also do not know the precise localization of event. To find the precise localization of event, we temporally cut a video into multiple segments and find the segments corresponding to the event. Thus event localization can be considered as a typical multiple instance learning problem, where each segment is an instance and all segments of a

video clip compose a bag. By this formulation, event boundaries can be located with the selection of more better segments.

In this paper, we propose a generic framework to automatically detect events from three realistic and challenging video datasets: sports, movie and news. We try to aggregate two sources of video data from text analysis and web video search together under a semi-supervised learning framework. To gain the localization of event in video data, multiple instance learning is adopted. Therefore, we formulate event detection into a Graph-based Semi-Supervised Multiple Instance Learning problem. Compared with the existing approaches, the contributions of our work can be summarized as follows.

- We present a generic framework for event detection in variant video genres. By combination of multi-modality information in video data, the generic event models are constructed based on video content and suitable for various video event analysis.
- To obtain the effective event models in variant video genres, we design a **Graph-based Semi-Supervised Multiple Instance Learning (GSSMIL)** method to integrate expert labeled data obtained by text analysis and unlabeled data collected from Internet, which improves the detection performance and solves the insufficient training data problem by resorting to Internet data source.
- To construct the discriminative graph for event model training, we introduce a **Multiple Instance Learning Induced Similarity (MILIS)** measure. The learned similarity considers the class structure which is ignored by the existing similarity measures [5, 7].

The rest of the paper is organized as follows. Section 2 reviews the related work. The framework of the proposed approach is described in Section 3. The technical details of video collection and graph-based label propagation are presented in Section 4 and 5, respectively. Experimental results are reported in Section 6. We conclude the paper with future work in Section 7.

2. RELATED WORK

The most related work to our method is event detection, semi-supervised learning and multiple instance learning, we review the state-of-the-arts of these three topics, respectively.

2.1 Event Detection

Most of existing work in event detection focuses on one type of video, such as movie, sports or news. For event detection in movies, much work [23, 8, 12] incorporates visual information, closed-captioned text, and movie scripts to automatically annotate videos in movies for classification, retrieval and annotation of videos. For event detection in sports video, most of the previous work is based on audio/visual/textual features directly extracted from video content [27, 13, 35, 31]. These approaches heavily rely on audio/visual/textual features directly extracted from the video content itself. Some work uses text information [33, 26] such as close caption and web text for event analysis. Similar to the event detection in sports video, there is a lot of work for event analysis in news video [22, 19, 16]. By exploiting the available audio, visual and closed-caption cues, the semantically meaningful highlights in a news video are located and event boundaries are extracted.

Analysis and modelling of abnormal event detection for video surveillance has also been studied [36, 4, 24]. These methods can be broadly categorized according to the type of scene representation. One very popular category is based on trajectory modeling [36, 4], and the other is based on motion and appearance representations [24]. However, these approaches are unsuitable for multiple events detection in complex videos. Most of existing work uses domain knowledge [23, 33], which is difficult to be used for other video domains. For example, the methods used for event detection in movie cannot be applied to other domains such as sports videos that do not provide associated scripts; Event detection approaches in sports video using text analysis [33] cannot be applied to many videos without text information.

Different from the precious work, we propose a generic framework for event detection in different video domains. The proposed method uses the videos with text information to learn model and then propagate labels to those videos with or without text information. Our learned model is based on video content. Therefore, it can be used for more generic video event analysis.

2.2 Graph-based Semi-supervised Learning

In the past few years, the graph-based semi-supervised learning approach has attracted a lot of attention due to its elegant mathematical formulation and effectiveness in combining labeled and unlabeled data through label propagation [21, 28]. The weight of the edge is the core component of a graph, which is crucial to the performance of the semi-supervised learning. The popular methods for the weight assignment include K-Nearest Neighbor (KNN), Gaussian Kernel Similarity (GKS) [5] and Sparsity Induced Similarity measure (SIS) [7] based on sparse decomposition in L_1 norm sense. The main drawback with these approaches is that their performance is sensitive to the parameter variation and they do not take the label information into consideration. Different from the previous methods, we propose a new approach to measure the similarities based on class structure information.

2.3 Multiple Instance Learning

There is little work [18] to detect events with the multiple instance representations [20, 30], which is most suitable for event detection in the videos. For multiple instance representations, each segment of a video clip is an instance and all segments of a video clip is a bag. Labels (or events) are attached to the bags while the labels of instances are hidden. The bag label is related to the hidden labels of the instances as follows: the bag is labeled as positive if any instance in it is positive, otherwise it is labeled as negative. For our task, it is effective to distinguish a positive event instance (i.e., the particular event) from a negative instance (i.e, the background frames) with the help of multiple instance representations.

3. FRAMEWORK OF OUR APPROACH

The framework of our proposed approach is illustrated in Fig.1. It contains three primary parts: video collection, GSSMIL and event detection. For video collection, it includes expert labeled data collection, event-relevant data collection and de-noising. For expert labeled data collection, text information of different kinds of video is used to structure the video segment and detect the start and end boundaries of the event to get a video event clip. Based on overlapped segmentation and de-noising of these video clips, a small-scale expert labeled database is collected. By querying key words from the web, we obtain some raw videos. After segmentation and de-noising, a large-scale event-relevant database is constructed. In the processing of de-noising, a Bayesian rule to efficiently remove some noise is adopted. For the GSSMIL module, we try to combine the expert labeled data and unlabeled data (from event-relevant data) for event detection. To effectively obtain

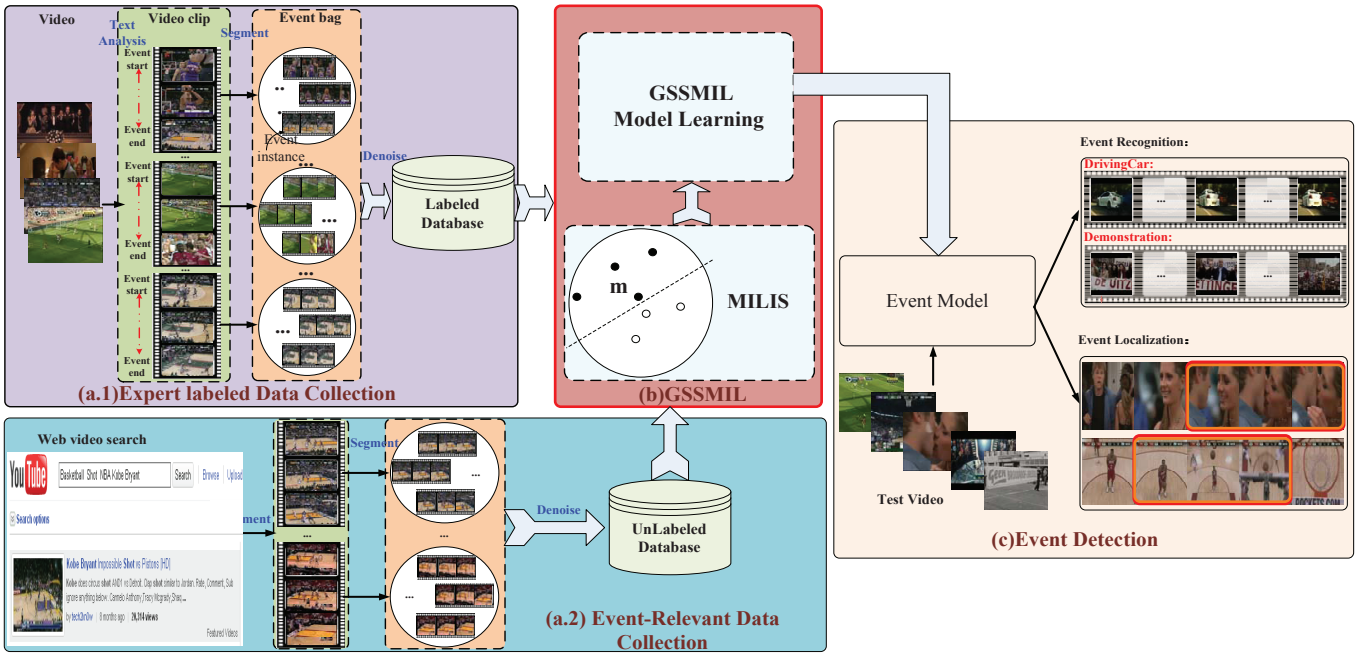


Figure 1: The proposed framework. For better viewing, please see the original color pdf file.

the similarity measure for the graph construction, we present an MILIS measure by considering the class structure. Finally, based on the learned event model, event recognition and localization are realized. The proposed approach is evaluated on highly challenging data from different video domains: movie, sports and news, and the experimental results are encouraging. The technical detail of each module in the framework will be described in the following sections.

4. VIDEO COLLECTION

In this section, we introduce how to collect annotated video samples and construct three different video datasets. To avoid manually labeling a large amount of video data, we design a smart strategy to automatically collect videos from professional broadcast service providers and Internet. The small part of labeled videos are obtained by the analysis and alignment of well-structured video related text (e.g. movie scripts, web-casting text, close caption), while the large part of event-relevant videos are collected from the Internet by querying related events from the video search engine (e.g. YouTube) and filtering noise.

4.1 Expert Labeled Data by Text Analysis

Here, we introduce an automatic procedure, as shown in Fig. 1(a.1), for collecting videos from multiple video types (sports, movie and news) supported by professional broadcast service providers. For different data sources, there are different available text information. For movie videos, we follow [14, 23, 12] using supervised text classification to detect events to automatically collect training samples. The OpenNLP toolbox [17] for natural language processing and part of speech (POS) tagging to identify instances of nouns, verbs and particles are applied to avoid manual text annotation. In addition, we also use named entity recognition (NER) to identify people's names. Based on results of POS and NER, we search for patterns corresponding to particular classes of events. Scripts describe events and their order in video but usually do not provide time information. We find temporal localization of dialogues in scripts by matching script text with the corresponding subtitles using dynamic programming. Then we estimate temporal localizations of events by transferring time information from subtitles to scripts. Using this procedure, a set of short video clips (segmented by subtitle timestamps) with corresponding script parts, i.e., textual descriptions, are obtained. For sports videos, we use web-casting text, which is usually available online and provided freely by almost all broadcasters [33, 9]. Keywords by which events are labeled are first predefined, then the time stamps where events happen are extracted from well defined syntax structure syntax web-casting texts by using the keywords as input query key to a commercial software, dtSearch [32]. This software is a sophisticated text search engine that has the ability to look-up word features dealing with the effect of fuzzy, phonic, wildcard, stemming and thesaurus search options. Similar to sports videos, the methods [22, 16] are adopted to find the temporal localizations of events with the closed-caption for news videos.

4.2 Event-Relevant Data from Internet

By using text information, we can obtain expert labeled data, however, it is still very difficult to handle text analysis and time-alignment to collect enough labeled video data. Moreover, there are many videos without their corresponding text information. The Internet is a rich source of information, with many event videos taken under various conditions, which are roughly annotated. It is convenient to use such a collection of videos as event-relevant data to improve the performance of event detection. By doing this, our work tries to fuse two lines of research "Internet vision" and "event detection" together and and improve event detection performance.

We query the event labels on a web video search engine like YouTube or Google. Based on the assumption that the set of retrieved videos contains relevant videos of the queried event, we can construct a large-scale video dataset, which includes videos taken from multiple viewpoints in a range of environments. The chal-

lenge is how to use these videos, because content in Internet is very diverse, which leads to the retrieved videos with much noise. For example, for a "Basketball Shot" query, a search engine is likely to retrieve some introduction videos of basketball shot. Our method must perform well in the presence of such noise. In this work, we adopt multiple keywords search ("Basketball Shot NBA Kobe Bryant") and propose an efficient method to remove some noise from the dataset in Section 4.3. Compared with data obtained by text analysis, we call this collection as event-relevant dataset. In our semi-supervised learning algorithm, this dataset is used as unlabeled data.

4.3 Segmentation and De-noising by Bayesian Rule

By text analysis and Internet, we can get a small-scale expert labeled data and a large-scale event-relevant video data. However, the labeled data only have the weakly associated labels, which means we know the video's label, but there is no precise information about the localization of event in video. For the event-relevant data, we also do not know the precise localization of event. To solve this problem, we perform temporal segmentation of video clips and get segments composed of contiguous frames. Our target is to jointly segment video clips containing a particular event, that is, we aim at separating what is shared within the video clips (i.e., the particular event) from what is different among these (i.e, the background frames). Given a video clip v_i containing the event of interest but at unknown position within the clip, the clip v_i is represented by n_i temporally overlapping segments centered at frames $1, \ldots, n_i$ represented by histograms $h_i[1], \ldots, h_i[n_i]$. Each histogram captures the l_1-normalized frequency counts of quantized space-time interest points and audio features, as described in section 6.2.

Let v_i^+ denote a positive video clip and v_i^- denote a negative video clip. v_{ij}^+ is the j^{th} segment of a positive video clip v_i^+ and v_{ij}^- denotes the j^{th} segment of a negative video clip v_i^-. Let $\{v_1^+, v_2^+, \ldots, v_m^+, v_1^-, v_2^-, \ldots v_n^-\}$ denote the set of m positive and n negative training video clips obtained by text analysis. $l(v_i) \in \{+1, -1\}$ is the bag label of v_i and $l(v_{ij}) \in \{+1, -1\}$ is the instance label of v_{ij}. For the negative video clips, all their segments are negative. However, for the positive video clips, their all segments must contain at least one true positive segment, and they may also contain many negative segments due to much noise, and imprecise localizations. The goal of de-noising is to identify the true positive segments in the positive video clips and remove some negative segments.

We assume that given a true positive segment s, the probability that a segment v_{ij} is positive is calculated as follows:

$$\Pr(l(v_{ij}) = +1|s) = \exp(-\frac{\|s - v_{ij}\|^2}{\delta_s^2}) \tag{1}$$

where $\|\bullet\|$ represents L2-norm, and δ_s is a parameter learned from the training data.

Given a true positive segment s, the probability that a video clip v_i is a positive video clip is defined as follows:

$$\begin{aligned}\Pr(l(v_i) = +1|s) &= \max_{v_{ij} \in v_i} \Pr(l(v_{ij}) = +1|s) \\ &= \max_{v_{ij} \in v_i} \exp(-\frac{\|s - v_{ij}\|^2}{\delta_s^2}) = \exp(-\frac{d^2(s, v_i)}{\delta_s^2}),\end{aligned} \tag{2}$$

where $d(s, v_i) = \min_{v_{ij} \in v_i} \|s - v_{ij}\|$. In other words, the distance $d(s, v_i)$ between a segment s and all segments of a video clip v_i is simply equal to the distance between s and the nearest segment of v_i. Then $\Pr(l(v_i) = +1|s) - \Pr(l(v_i) = -1|s) = 2\exp(-\frac{d^2(s,v_i)}{\delta_s^2}) - 1$. If $\Pr(l(v_i) = +1|s) \geqslant \Pr(l(v_i) = -1|s)$, we get $d(s, v_i) \leqslant \delta_s\sqrt{\ln 2}$. For a negative segment (i.e., false positive segment), however, its distances to the positive and negative video clips do not exhibit the same distribution as those from s. Since some positive video clips may also contain negative segments just like the negative video clips, the distances from the negative segment to the positive video clips may be as random as those to the negative video clips. This distributional difference provides an informative hint for identifying the true positive segments. Therefore, given a true positive segment s , there exists a threshold θ_s which allows the decision function defined in Eq.(3) to label the video clips according to the Bayes decision rule.

$$h_{\theta_s}^s(v_i) = \begin{cases} +1 & \text{if } d(s, v_i) \leqslant \theta_s \\ -1 & \text{otherwise,} \end{cases} \tag{3}$$

where $\theta_s = \delta_s\sqrt{\ln 2}$ determined by training data as follows:

$$P(s) = \max_{\theta_s} P_s(\theta_s), \tag{4}$$

where $P_s(\theta_s)$ is an empirical precision and defined as follows:

$$P_s(\theta_s) = \frac{1}{m+n} \sum_{i=1}^{m+n} \frac{1 + h_{\theta_s}^s(v_i) l(v_i)}{2}. \tag{5}$$

In this way, for each segment from the labeled dataset, we can obtain the P_s. Based on this value, we can remove some segments of each video clip. Note that the exact number of true positive segments for one specific positive video clip is unknown. To handle this problem, we propose that for a video clip: if $P(s) > th_1$, s is selected, where th_1 is a threshold and is manually set to be 0.5 in our experiments. Based on our experiments, this method is able to well solve our problem. In this way, we can remove irrelevant segments and video clips obtained by text analysis and construct an expert labeled dataset.

For the data obtained by web video search, we can also de-noise them by using the expert labeled data. Because data obtained from web have more noise than data obtained by text analysis, we set th_1 as 0.8 to obtain much cleaner data. Moreover, we adopt another strategy to confirm the reliability of the selected data from the web by the classifier introduced in Section 5.3. We can get segments for each video clip. Then, each segment s is classified using all classifiers trained with labeled data and has mean score Sc_s. If the score $Sc_s > th_2$, this segment is selected and the video clip is viewed as event-relevant data. If scores of all segments of a video clip are below th_2, the video clip is not selected. In our experiments, the th_2 is manually set to be 0.7. After de-noising by the two strategies, we collect a large-scale of more cleaner event-relevant data from web, and this data will be used as unlabeled data to give distributive information.

5. GRAPH-BASED SEMI-SUPERVISED MULTIPLE INSTANCE LEARNING (GSSMIL)

In this section, we introduce the GSSMIL algorithm which combines labeled data (as introduced in Sec. 4.1) and unlabeled data (as described in Sec. 4.2) and adopts multiple instance learning to detect the positive event instances from the event bags, where we consider event with precise localization in a video clip as positive event instance and event with imprecise localization as negative event instance, and the corresponding video clip is viewed as an event bag. Next, we introduce the problem description in Section 5.1, and how

to construct the graph and learn the similarity measure for the graph are presented in Section 5.2 and Section 5.3, respectively. Finally, we introduce how to solve the objective function in Section 5.4.

5.1 Problem Description

After video collection in Section 4, each segment of a video clip is viewed as an instance (**event instance**), and all segments of the video clip comprise of a bag (**event bag**). We use the following notation throughout this paper. Let $L = \{(x_1, y_1), \ldots, (x_{|L|}, y_{|L|})\}$ be the labeled data and let $U = \{x_{|L|+1}, \ldots, x_{|L|+|U|}\}$ be the unlabeled data. Each bag x_b is a set of instances $\{x_{b,1}, x_{b,2}, \ldots, x_{b,n_b}\}$, with its label denoted by $y_b \in \{0, +1\}$, where $+1$ is used for positive bag and 0 for negative bag. b is the index of bag, and n_b is the number of all instances of bag x_b. Each $x_{b,j} \in R^d$ is a d-dimensional feature vector representing an instance. Without loss of generality, we assume that the first L_1 bags are positive and the following L_2 bags are negative ($|L| = L_1 + L_2$). To describe the relationship between bags and instances, we use $x_{b,j} \in x_b$ to represent that $x_{b,j}$ is an instance from bag x_b and its label is represented as $y_{b,j}$.

Our task is to learn a soft label function $\hat{\boldsymbol{f}} : R^d \rightarrow [0, +1]$ that learns the label for each instance, we denote the predicted soft label of instance j by $f_j = \hat{\boldsymbol{f}}(x_{bj})$. Then, the labels of the bags can be calculated. We define a bag x_b's label f_b^* to be determined by the largest value of its corresponding instances' soft labels:

$$f_b^* = \max_{j, x_{b,j} \in x_b} f_j. \tag{6}$$

5.2 Graph construction

In this section, we formulate the graph-based semi-supervised multiple instance learning in an instance-level way and define the cost criterion based on instance labels. Consider a graph $G = (V, E)$ with nodes corresponding to N feature vectors. There is an edge for every pair of the nodes. We assume that there is an $N \times N$ symmetric weight matrix $W = [w_{ij}]$ on the edges of the graph, where N is the number of all instances. The weight for each edge indicates the similarity between the two nodes that are connected by the edge. Intuitively, similar unlabeled samples should have similar labels. Thus, the label propagation can be formulated as minimizing the quadratic energy function [37]:

$$E_1(f) = \frac{1}{2} \sum_{i,j} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2, \tag{7}$$

where d_i is the sum of the i^{th} row of W and denote $D = diag(d_1, \ldots, d_N)$. f_i is the label of instance i, and f_i should be nonnegative. Assume the instance i is the i^{th} instance of event bag b, f_i can be denoted as $f_i = \left[f_{bi}^1, \cdots f_{bi}^c, \cdots f_{bi}^C \right]$. We can obtain some prior knowledge from the labeled bag to its instance label, that is, f_{bi}^c must be 0 if the bag x_b does not contain label c. In this way, bag label information is applied.

The Eq.(7) just controls the complexity in the intrinsic geometry of the data distribution and the smoothness of label over the instance-level graph. For our problem, we need to consider the constraints based on labeled bags, For a negative bag, it is straightforward to see that all instances in the bag are negative, i.e., $f_j = 0$, for all $j : x_{b,j} \in x_b$. Thus we have the penalty term:

$$E_2(f) = \sum_{b=1+L_1}^{|L|} \sum_{j: x_{b,j} \in x_b} f_j. \tag{8}$$

Meanwhile, for a positive bag, the case is more complex because a positive bag may contain negative instances as well. Actually, only one positive instance is necessary to determine a positive bag. Thus, we define the penalty term for a positive bag to be only related to the instance with the largest soft label:

$$E_3(f) = \sum_{b=1}^{L_1} \left(1 - \max_{j: x_{b,j} \in x_b} f_j \right). \tag{9}$$

By combing the three items, we have the following cost criterion:

$$\begin{aligned} E(f) = \frac{1}{2} \sum_{i,j} w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 + \alpha_1 \sum_{b=1+L_1}^{|L|} \sum_{j: x_{b,j} \in x_b} f_j \\ + \alpha_2 \sum_{b=1}^{L_1} \left(1 - \max_{j: x_{b,j} \in x_b} f_j \right), \end{aligned} \tag{10}$$

where α_1 and α_2 are two parameters used to balance the weight. In our experiments, we set $\alpha_1 = \alpha_2 = 10$, and obtained a good performance.

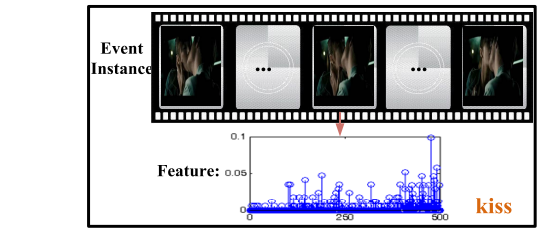
Once we have found the optimal labels of the instances by minimizing the cost criterion $E(f)$, the bag-level label of any bag x_b can be calculated by taking the maximum value of its instances' labels using Eq.(6). The only two problems left are how to obtain efficient similarity measure $W = [w_{ij}]$ and how to solve the optimization task in Eq.(10). For the first problem, we propose a multiple instance learning based method to learn the similarity measure $W = [w_{ij}]$ and introduce this in Section 5.3. Due to the existence of the $max(\bullet)$ function in the loss function Eq.(10) for positive bags, $E(f)$ is generally non-convex, and cannot be directly optimized. In section 5.4, we will derive a sub-optimum solution to this problem.

5.3 Multiple Instance Learning Induced Similarity (MILIS) Measure

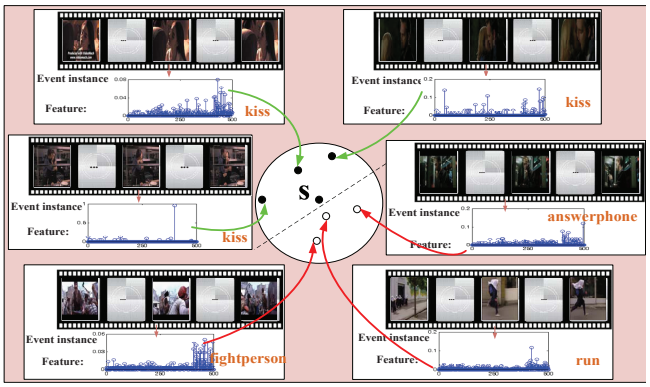
One main drawback of most existing similarity measures, such as the Euclidean distance and Gaussian Kernel Similarity measure, is that the similarity measurement completely ignores the class structure. For example, in Fig. 2, given an event instance s belonging to category c, it is possible that some event instances from the same class to s are less similar than the ones from other classes when a predefined and heuristic distance metric is adopted. To tackle this problem, we attempt a discriminative solution to get truly similarity by learning some classifiers. Here, we formulate the similarity measure learning as a problem of Multiple Instance Learning (MIL) [10] and mi-SVM [2] is employed to solve the problem.

Next, we will introduce how to train a classifier for an event instance s from the category c. This training process can be repeated for all classifiers of different kinds of event instances. For the event instance s denoted as feature vector I_s, its classifier is trained in the hyper-sphere centered at I_s with radius of r_s in the feature space (as showed in Fig. 2). The training samples are the samples in class c denoted as positive bags and those in other categories denoted as negative ones. This strategy filters out the instances that are very different from s for each bag and enables the classifier to be learned only in the local feature space. Therefore, it is very efficient to reduce the computational burden and learn a discriminative classifier.

Define the distance from event instance I_s to event bag x_b as $d_{b,j,s} = \min_j \|I_s - x_{b,j}\|$, where $\|\bullet\|$ represents L2-norm. In practice, it is found quite robust and in majority cases the positive instance in the positive bag x_b is $\left\{ x_{b,j^*} | j^* = \arg\min_j \|x_{b,j} - I_s\| \right\}$. Based on this observation, $r_s$is set as follows: $r_s = \underset{b \in pos}{mean}(d_{b,j,s}) +$



(a)



(b)

Figure 2: Learning similar event instances to event instance s. (a) event instance s, which is one instance of kissing event bag. (b) Learning a classifier to describe the similarities of other event instances to s. "• "represents similar event instances from the same kind of event with s, and "○ "represents unrelated event instances.

$\beta \times \underset{b \in pos}{std}\,(d_{b,j,s})$, where β is a trade-off between efficiency and accuracy. The larger β, the less probability that one positive instance will be filtered out before training, and the more event bags will be involved during solving the mi-SVM. In our experiments, the β is manually set to be 2.5 by experience.

It is observed in experiments that there may be too many event instances falling into the hyper-sphere. To be more efficient, within the hyper-sphere, at most k_p nearest event instances to I_s are selected for each positive event bag, and k_n for each negative event bag. In our experiments, $k_p = 5$ and $k_n = 2$ are used by experience. Experiments show that this strategy can significantly reduce the computational burden.

Denote $y_{b,j}$ to be the instance label of event instance $x_{b,j}$ and y_b the label of event bag x_b, where $x_{b,j}$ is the feature of the event instance j in the event bag x_b. mi-SVM is formulated as follows:

$$\begin{aligned} &\min_{\{y_{b,j}\}} \min_{w^*,b_0,\xi} \frac{1}{2}\|w^*\|^2 + C\sum_{x_{b,j}} \xi_{x_{b,j}} \\ &s.t. \sum_j \frac{y_{b,j}+1}{2} \geqslant 1,\ \forall x_b\ s.t.\ y_b = 1 \\ &\quad y_{b,j} = -1,\ \ \forall x_b\ s.t.\ y_b = -1 \\ &\quad \forall j : y_{b,j}(\langle w^*, x_{b,j}\rangle + b_0) \geqslant 1 - \xi_{x_{b,j}}, \xi_{x_{b,j}} \geqslant 0, \\ &\quad\quad y_{b,j} \in \{-1, 1\}. \end{aligned} \tag{11}$$

Denote $mi - SVM_s$ as the trained classifier corresponding to event instance s. Based on this classifier, all event instances can be projected to real value with a function. For simplicity, the project function is defined as follows:

$$g_s(x_{b,j}) = \begin{cases} mi - SVM_s(x_{b,j})\ \exists x_{b,j}, s.t.\ \|x_{b,j} - I_s\| \leqslant r_s \\ 0 \quad otherwise \end{cases}, \tag{12}$$

where $mi - SVM_s(x_{b,j}) \in \mathbb{R}$ is the output of the classifier $mi - SVM_s$ with the input $x_{b,j}$. Based on this score, the similarity between instance s and $x_{b,j}$ can be simply defined as follows:

$$w_{sj} = \begin{cases} g_s(x_{b,j}) & if\ \ g_s(x_{b,j}) > 0 \\ 0 & otherwise \end{cases} \tag{13}$$

Therefore, for each event instance s in labeled data set, we can get its corresponding classifier and the similarities with other event instances. Though there may be some instances in positive event bags belonging to negative instances after de-noising. It is still efficient to learn the similarity measure based on our experimental results.

Based on the L labeled data and U unlabeled data, the similarity measure W can be splitted into labeled and unlabeled sub-matrices: $W = \begin{pmatrix} W_{LL} & W_{LU} \\ W_{UL} & W_{UU} \end{pmatrix}$, where $W_{LU} = W_{UL}$. W_{LL} and W_{LU} can be obtained by Eq.(13) using learned classifiers. For the unlabeled data, we adopt Euclidean distance to measure the similarity W_{UU} between data points.

5.4 Iterative Solution Using CCCP

Because $E_3(f)$ defined by Eq. (9) is non-convex, $E(f)$ can be viewed as that a convex function adds a concave function. Therefore, we adopt the constrained concave convex procedure (CCCP) to find the sub-optimum solution. CCCP is proposed in [29] as an extension of [34], and is theoretically guaranteed to converge. It works in an iterative way: at each iteration, the first order Taylor expansion is used to approximate the non-convex functions, and the problem is thus approximated by a convex optimization problem. The sub-optimum solution is given by iteratively optimizing the convex subproblem until convergence.

Note that $\max(\bullet)$ is not differentiable at all points. To use CCCP, we have to replace the gradients by the subgradients. Let $l = [f_{b1}^c, \cdots, f_{bj}^c, \cdots, f_{bn_b}^c]^T$, where f_{bj}^c denotes the probability that the j^{th} instance in the b^{th} bag belongs to the c^{th} class and $c \in \{1 \cdots C\}$, C is the total number of classes and n_b is the number of instances in the b^{th} bag. We pick the subgradient with ρ, which is an $n_b \times 1$ vector and its j^{th} element is given by

$$\rho_j = \begin{cases} \frac{1}{\tau} & \text{if } {l_j}^{(t)} = \max({l_b}^{(t)}) \\ 0 & \text{otherwise} \end{cases}, \tag{14}$$

where $\max(l_b^{(t)})$ represents the largest label value of bag b and τ is the number of instances with the label value $\max(l_b^{(t)})$. At the $(t+1)^{th}$ iteration, we estimate the current l based on $l^{(t)}$ and the corresponding ρ_j. As $\rho^T l^{(t)} = \sum_j \rho_j l_j^{(t)} = \max l^{(t)} \sum_{\rho_j \neq 0} \rho_j = \max l^{(t)}$, for the function $\max(l)$, its 1-st order Taylor expansion is approximated as $(\max l)_{l^{(t)}} \approx \rho^T l$.

For the t-th iteration of CCCP, the objective function in Eq.(10) is rewritten in matrix form as follows:

$$\begin{aligned} &\min_F\ Tr(F^T L F) + \alpha_1 \sum_b \sum_c (1 - Y_{bc}) h_c F^T q_b + \\ &\quad \alpha_2 \sum_b \sum_c Y_{bc}(1 - h_c \beta U_b F h_c^T) \\ &s.t.\ F \geqslant 0,\ Fe_1 = e_2 \end{aligned} \tag{15}$$

where L is a Laplace matrix $L = D - W$, with D being the degree matrix and $Tr(\cdot)$ represents matrix *trace* operator, $F = [f_{11}, \cdots, f_{1n_1}, \cdots, f_{Bn_1}, \cdots, f_{Bn_B}]^T$ and $F \in R^{N \times C}$. B is the number of all bags and $N = \sum_{b=1}^{B} n_b$ is the number of all in-

(a) Movie Video (b) Sports Video (c) News Video

Figure 3: Realistic samples for three different video types: movie video, sports video and news video and their corresponding events. All samples have been automatically retrieved by text analysis.

stances. Each row of F corresponds to the posterior probability distribution of an instance, hence should be: (1) positive (2) l_1 normalized. Therefore, the constraint of Eq.(15) is necessary. $Y = [Y_{bc}]$, $Y_{bc} = 1$ if bag b belongs to the c^{th} class, otherwise $Y_{bc} = 0$. h_c is a $1 \times C$ indicator vector, the c^{th} element of which is one and others are zero, and $q_b = [\underbrace{0,\cdots,0}_{1,\cdots,b-1}, \underbrace{1,\cdots,1}_{b}, \underbrace{0,\cdots,0}_{b+1,\cdots,B}]^T$ is an $N \times 1$ vector whose all elements, except for those elements corresponding to the b^{th} bag, are zeros. β is a $C \times N$ matrix, $\beta = [\beta_1, \cdots, \beta_b, \cdots, \beta_B]$, each $\beta_b = [\beta_{b1}^T, \cdots, \beta_{bc}^T, \cdots, \beta_{bC}^T]^T$ is a $C \times n_b$ matrix corresponding to bag b and $\beta_{bc} = \eta^T$. $e_1 = \mathbf{1}_{C\times 1}$ and $e_2 = \mathbf{1}_{N\times 1}$ are both all-one vectors, α_1 and α_2 are two parameters used to balance the weight. $U_b = diag\,(u_1, \cdots, u_b, \cdots u_B)$ is an $N \times N$ diagonal block matrix, where $u_k = 0_{n_k \times n_k}$ for $k = 1, \cdots, b-1, b+1, \cdots B$ and $u_b = I_{n_b \times n_b}$, I represents an identity matrix.

The subproblem in Eq. (15) is a standard quadratic programming (QP) [6] problem and can be solved by any state-of-the-art QP solvers. In our work, it is solved efficiently with global optimum using existing convex optimization packages, such as Mosek [1]. Running CCCP iteratively until convergence, we can obtain the sub-optimum solution for the instance labels. The label for each bag is then calculated as the largest label of all its instances using Eq. (6).

Out-of-sample Extension: For a new testing instance t, its label is given as:

$$f_t = \sqrt{d_t} \sum_j w(j,t) \frac{f_j}{\sqrt{d_j}} \Big/ \sum_j w(j,t), \qquad (16)$$

where $w(j,t)$ represents the similarity between instance j and t. d_j and d_t have the same meaning as in Eq.(7). d_t is an unknown constant for a particular testing instance t, and we can ignore it when making decision. f_j is the obtained instance label by Eq.(10).

When given a testing bag with multiple instances, we make use of Eq.(16) to obtain instance label and then apply Eq.(6) to classify the bag. Because we can obtain a label for each instance of a bag, the localization of event can be also obtained. Here, Gaussian kernel based temporal filtering is conducted to smooth the event instances from a video stream taking account of the temporal consistency of events.

6. EXPERIMENTAL RESULTS

In this section, we present extensive experimental results on movie, sports and news video datasets in order to validate the proposed approach.

6.1 Dataset Introduction

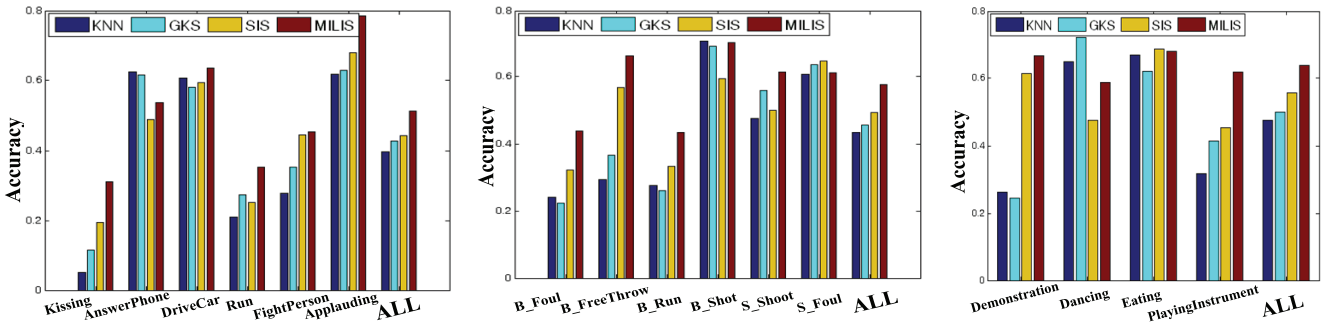
Because there is little work to handle different kinds of video, no publicly generic dataset is avaliable. By using the method introduced in Section 4, we obtain three different video datasets. For movie videos, we select 6 different kinds of representative events: AnswerPhone, DriveCar, Kissing, FightPerson, Run and Applauding. 981 unlabeled data are obtained from the web and 311 expert labeled data are obtained by text-video alignment. The dataset is from about 60-hour videos. For sports videos, we select 6 different kinds of events: Basketball Foul (B_Foul), Basketball Free Throw (B_FreeThrow), Basketball Run (B_Run), Basketball Shot (B_Shot), Soccer Foul (S_Foul) and Soccer Shoot (S_Shoot). 913 unlabeled data are obtained from the web and 316 expert labeled data are obtained by text-video alignment. There are about 25-hour videos for this dataset. For news videos, we select 4 kinds of events: Eating, PlayingInstrument, Demonstration and Dancing. 863 unlabeled data are obtained from the web and 311 expert labeled data are collected by text-video alignment from about 20-hour videos. The number of test data is 253, 211 and 203 for the three datasets, respectively. Note that the three different kinds of videos are very challenging for video analysis due to its loose and dynamic structure as shown in Fig. 3.

6.2 Video Feature Extraction

Visual features and audio features are complimentary and important for video event detection. For example, Basketball Free Throw and Basketball Shot are most similar just using visual features, however, audio features, such as whistle of referee, are discriminative. To the contrary, visual features are very important to distinguish dancing and playing a guitar, because they both have similar background music and different motion features. In the following subsections, we will introduce two features, respectively.

6.2.1 Spatio-temporal Features

Sparse space-time features have recently shown good performance for video analysis [11]. They provide a compact video representation and tolerance to background clutter, occlusions and scale changes. We detect interest points using a space-time extension of the Harris operator. To characterize motion and appearance of local features, we compute histogram descriptors of space-time volumes in the neighborhood of detected points. For a 3D video patch in the neighborhood of each detected space-time interest point (STIP), it is partitioned into a grid with $3\times3\times2$ spatio-temporal blocks; 4-bin HOG descriptor and 5-bin HOF descriptor are then computed for all blocks and are concatenated into a 72-element and 90-element descriptors, respectively. The details can be found in [23]. For each volume we compute coarse histograms of oriented gradient (HoG)



(a) Movie Video Dataset (b) Sport Video Dataset (c) News Video Dataset

Figure 4: The accuracy of different similarity measures on the three datasets, respectively.

and optical flow (HoF). Normalized histograms are concatenated into HoG and HoF descriptor vectors and are similar in spirit to the well known SIFT descriptor. HoF is based on local histograms of optical flow. It describes the motion in a local region. HoG is a 3D histogram of 2D (spatial) gradient orientations. It describes the static appearance over space and time. Then the two descriptors are concatenated into one 162-dimensional vector, which is reduced by PCA to 60.

6.2.2 *Audio Features*

The mel-frequency cepstral coefficients (MFCCs) [25] are proved more efficient [38] for audio recognition. Therefore, we adopt the MFCCs to represent audio. The MFCCs are based on a short-term spectrum, where Fourier basis audio signals are decomposed into a superposition of a finite number of sinusoids. The power spectrum bins are grouped and smoothed according to the perceptually motivated Mel-frequency scaling. Then the spectrum is segmented by means of a filter bank that typically consists of overlapping triangular filters. Finally, a discrete cosine transform applied to the logarithm of the filter bank outputs results in vectors of decorrelated MFCC features. In our experiments, we use 13-dimensional MFCC features.

6.2.3 *Bag of Features*

For the two different modality features, we build bag-of-features (BoF) respectively. This requires the construction of visual vocabulary. In our experiments we cluster a subset of $400k$ features sampled from the training videos with the k-means algorithm for visual features. The number of clusters is set to $k = 400$ for visual features and $k = 100$ for audio features with a subset of $100k$ features sampled from the training data, which have shown empirically to give good results. The BoF representation then assigns each feature to the closest (we use Euclidean distance) vocabulary word and computes the histogram of visual word occurrences over a space-time volume corresponding to the segments obtained by over segmentation of the entire video clip. Then, for each segment, the two histograms are concatenated into one 500 dimensional vector and then normalized.

Table 1: Comparison of the average accuracy of different similarity measures on the three datasets, respectively.

Method / Dataset	KNN	GKS	SIS	MILIS
Movie	39.76%	42.82%	44.23%	**51.27%**
News	47.48%	50.10%	55.79%	**63.87%**
Sports	43.35%	45.58%	49.41%	**57.71%**

6.3 Recognition Evaluation of Different Similarity Measurements

This experiment compares the proposed MILIS measure with three similarity measures, GKS, sparsity induced similarity measure(SIS) [7], and KNN on the three datasets. We compare our approach with the three methods, because they are very popularly and extensively used to measure similarity. For GKS, we use $d_{ij} = \exp(-\frac{\|p_i - p_j\|^2}{\delta^2})$ to measure similarity and the variance δ is set to be $1.5, 1.5, 1.2$ which achieved the best performance for the three datasets, respectively. For KNN, we use inner product similarity to find the K nearest neighbors while the number of nearest neighbors K is tuned by cross-validation. We found that 30, 30,20 work better for our experiments on the three datasets, respectively. Then, the similarity values between a sample and its K nearest neighbors are their correlation coefficients while those between the sample and the rest are set to 0. As for SIS, we normalize all feature vectors so that their L_2 norms are 1 before computing the weight matrix.

Fig. 4 shows the propagation accuracies of four different similarity measures: KNN, GKS, SIS, and MILIS. The x-axis is categories of different events. The y-axis is label propagation accuracy for individual classes and the mean accuracy for all of the categories denoted as "ALL". We can see that GKS (labeled as 'GKS') works better than KNN (labeled as 'KNN'), SIS (labeled as 'SIS') works better than GKS, and MILIS (labeled as 'MILIS') works the best. From Fig. 4, we notice that the mean accuracies for all of concepts of our method outperform the other methods on the three datasets respectively. The average precision values of different similarity measures on the three datasets are shown in Table 1. We can see that our method has an improvement of at least 7%. Fig. 4 also shows that there are some classes where our approach does not outperform the other approaches, such as AnswerPhone, Soccer Foul (S_Foul) and Dancing. To explain the reason, we give an example on news video dataset. We can see two similarity measures (GKS and KNN) outperform our method (MILIS) for "Dancing" recognition. This is because the four classes are very prone to be classified as "Dancing" using GKS and KNN with our video features, which leads to a high performance for "Dancing" recognition with a high false alarm rate. However, our proposed similarity measure considers the class structure, which improves the discrimination of the feature points and reduces the false alarm rate. Based on the mean accuracies for all of the concepts on the three datasets, we can confirm that our approach obtains the best performance compared among the existing methods. The result is obvious, because the existing similarity measures such as KNN, SIS and GKS completely ignore the class structure. However, our approach considers the distribution of feature points in the feature space and adopts a classifier to improve the similarity measure by using discriminative class information.

Table 2: Comparison of different learning strategy on the three datasets.

Method / Dataset	SVM	mi-SVM	GSSMIL
Movie	44.99%	47.85%	**51.27%**
News	53.23%	58.51%	**63.87%**
Sports	51.30%	53.31%	**57.71%**

6.4 Recognition Evaluation of Different Learning Strategies

Three experiments with different learning strategies are performed to validate the effectiveness of our method. First, we do not formulate the event detection into a multiple instance learning problem, instead, we use the entire video as a training sample and train an SVM classifier. The results are shown in Fig. 6 (labeled as 'SVM'). In the second experiment, we do not use the unlabeled data from Internet, and only employ the expert labeled data, its results are shown in Fig. 6 (labeled as 'mi-SVM'). The last experiment is our 'GSSMIL' method which formulates the event detection as multiple instance learning problem and combines labeled and unlabeled data under a semi-supervised framework.

The average precisions of three different learning strategies on the three datasets are shown in Table 2. From Table 2, we can see that the mean accuracies for all of concepts of our approach outperform the other methods on the three datasets, and have been improved about 4%. Fig. 6 also shows that certain results are worse for certain types of content, for example, AnswerPhone. This is because the five classes are very easy to be classified as this type of event, which leads to a high false alarm rate using 'SVM'. From the results of mean accuracies for all of the concepts, we can see that our GSSMIL performs better than the other methods. The reasons can be summarized as follows: (1) The entire video contains not only events of our interest, but also some clutter noises, which harms the classifier training and results in the bad performance. The increasing performance of GSSMIL clearly illustrates the importance of temporal event localization in the training data. In addition, it is very suitable to formulate the event analysis as a multiple instance learning problem. (2) By combining large-scale unlabeled data, the GSSMIL algorithm is effective to mine useful information.

6.5 Event Localization Evaluation in Video Sequence

In this section, we apply the GSSMIL algorithm described above to temporally localize event boundaries on the three video datasets. The quality of the segmentation is evaluated in terms of localization accuracy. The GSSMIL algorithm is evaluated on a set of 117, 114 and 105 events for movie, news and sports videos, respectively. Our testing and training videos do not share the same scenes or actors. For both the training and test set, the ground truth event boundaries were obtained manually.

The temporal localization accuracy is measured by the percentage of clips with relative temporal overlap to ground truth event segments greater than 0.3. This relatively loose threshold of 0.3 is used in order to compensate for the fact that temporal boundaries of events are somewhat ambiguous and not always accurately defined. Using this performance measure, we conduct the experiments for videos without text information alignment. For the news video dataset, the results are shown in Table 3. For video without text information, the GSSMIL correctly localizes 102 out of 114 clips, which corresponds to an accuracy of 61.6% (labeled as 'GSSMIL'). However, if we use mi-SVM, the precision is only 55.6% (labeled as 'miSVM'). The result shows that the localization performance is improved by using of unlabeled data. The average precision of event localization is shown in Table 4, which shows our algorithm can effectively localize event boundaries. Some automatically localized segments are shown in Fig. 5.

Table 3: Comparison of the performance of localization on news dataset.

Event	Demonstration	Dancing	Eating	PlayingInstrument
# Positive Sample	31	28	23	32
mi-SVM	51.6%	53.6%	60.8%	56.3%
GSSMIL	**61.2%**	**60.7%**	**65.2%**	**59.3%**

Table 4: Comparison of the average precision of localization on the three datasets.

Dataset	Movie	News	Sports
# Positive Sample	117	114	105
mi-SVM	43.6%	55.6%	47.6%
GSSMIL	**46.2%**	**61.6%**	**53.3%**

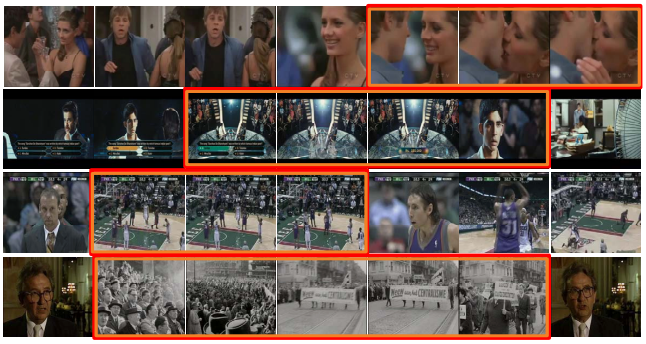
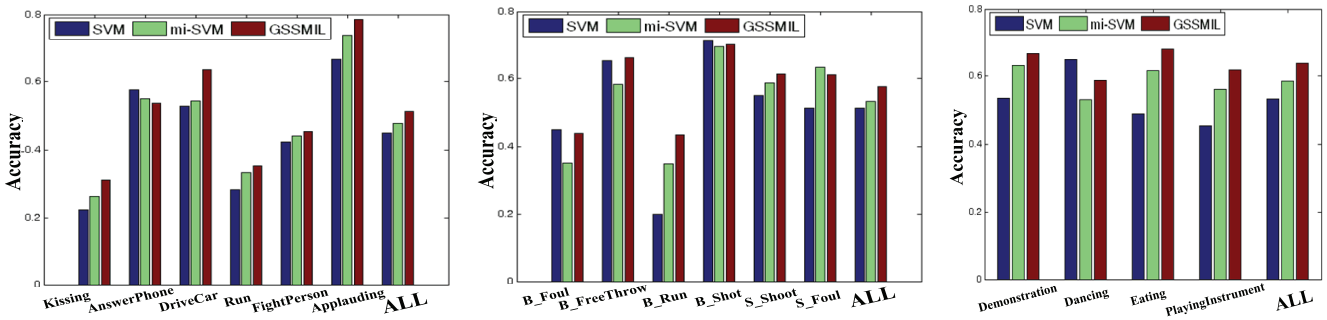


Figure 5: Some examples of temporally localizations of events by the proposed algorithm on the three datasets. Each row shows example frames from the entire video clip. Example frames of automatically localized events within the clips are shown in red. The four rows represent "Kissing" and "Applauding" on movie video dataset, "B_Shot" on sports video dataset, and "Demonstration" on news video dataset, respectively.

7. CONCLUSIONS

Video event detection is very important for content based video indexing and retrieval. This paper provides an effective event detection method using the GSSMIL strategy. To tackle the insufficient labeled data problem and alleviate human labeling effort, text information is mined and served as labels to assist model training. Besides the expert-level labels, Internet is also able to provide a huge amount of event-relevant data (used as unlabeled data). Our GSSMIL method can exploit both datasets together. To handle the ambiguity of event boundary in our labeled and unlabeled datasets, the GSSMIL algorithm incorporates a multiple instance learning module. Moreover, the GSSMIL algorithm employs an effective method to describe the sample affinity, which is proved to boost the event recognition and localization performance significantly. In the future, we will extend the generic method to more broad categories and more video domains. Moreover, we will combine the three datasets together and research the performance of our proposed method to detect all kinds of events.



(a) Movie Video Dataset (b) Sports Video Dataset (c) News Video Dataset

Figure 6: Comparison of different learning strategies. The result of an entire video as a training sample without no segmentation shown as 'SVM'; The result of just using labeled training data without unlabeled data is denoted as 'mi-SVM'; The result of our method is labeled as 'GSSMIL'.

8. ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (Project No. 60835002 and 90920303), and National Basic Research Program (973) of China (Project No. 2010CB327905).

9. REFERENCES

[1] http://www.mosek.com/.

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

[3] N. Babaguchi, Y. Kawai, and T. Kitahashi. Event based indexing of broadcasted sports video by intermodal collaboration, 2002. IEEE Transactions on Multimedia.

[4] A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection, 2008. In CVPR.

[5] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering, 2002. In NIPS.

[6] S. Boyd and L. Vandenberghe. Convex optimization, 2003. Cambridge University Press.

[7] H. Cheng, Z. Liu, and Z. Liu. Sparsity induced similarity measure for label propagation, 2009. In ICCV.

[8] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription, 2008. In ECCV.

[9] M.-S. Dao and N. Babaguchi. Sports event detection using temporal patterns mining and web-casting text, 2008. In AREA '08: Proceeding of the 1st ACM workshop on Analysis and retrieval of events/actions and workflows in video streams.

[10] T. Dietterich, R. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axis parallel rectangles, 1997. Artificial Intelligence.

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

[12] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video, 2009. In ICCV.

[13] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization, 2003. IEEE Trans. on Image Processing.

[14] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy - automatic naming of characters in tv video, 2006. In BMVC.

[15] M. Fleischman and D. Roy. Grounded language modeling for automatic speech recognition of sports video. In Proceedings of ACL-08: HLT.

[16] A. G. Hauptmann and M. J. Witbrock. Story segmentation and detection of commercials in broadcast news video, 1998. Advances in Digital Libraries.

[17] http://opennlp.sourceforge.net.

[18] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang. Action detection in complex scenes with spatial and temporal ambiguities, 2009. In ICCV.

[19] C. Huang, W. Hsu, and S. Chang. Automatic closed caption alignment based on speech recognition transcripts, 2003. Tech. Rep. 007, Columbia University.

[20] Y. Jia and C. Zhang. Instance-level semisupervised multiple instance learning, 2008. AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence.

[21] T. Kato, H. Kashima, and M. Sugiyama. Robust label propagation on multiple networks. 2009. IEEE TNN.

[22] J. G. Kim, H. S. Chang, K. Kang, M. Kim, J. Kim, and H. M. Kim. Summarization of news video and its description for content-based access. *International Journal of Imaging Systems and Technology*, 13:267–274, 2003.

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

[24] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes, 2010. In CVPR.

[25] M. Müller. Information retrieval for music and motion. *Springer*, page 65, 2007.

[26] B. N, K. Y, O. T, and K. T. Personalized abstraction of broadcasted american football video by highlight selection. *IEEE Trans Multimedia*, 6:575–586, 2004.

[27] Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for tv baseball programs, 2000. In Proc. of ACM Multimedia, Los Angeles.

[28] A. Singh, R. D. Nowak, and X. Zhu. Unlabeled data: now it helps, now it doesn't, 2008. In NIPS.

[29] A. Smola, S. Vishwanathan, and T. Hofmann. Kernel methods for missing variables, 2005. Proc. International Workshop on Artificial Intelligence and Statistics.

[30] C. Wang, L. Zhang, and H.-J. Zhang. Graph-based multiple-instance learning for object-based image retrieval, 2008. MIR '08: Proceeding of the 1st ACM international conference on Multimedia information retrieval.

[31] J. Wang, C. Xu, E. Chng, K. Wan, and Q. Tian. Automatic generation of personalized music sports video, 2005. In Proc. of ACM International Conference on Multimedia.

[32] www.dtSearch.com.

[33] C. Xu, J. Wang, K. Kwan, Y. Li, and L. Duan. Live sports event detection based on broadcast video and web-casting text, 2006. In MM'06 Conference Proceedings.

[34] A. Yuille and A. Rangarajan. The concave-convex procedure. 15(4):915–936, 2003. Neural Computation.

[35] D. Zhang and S. Chang. Event detection in baseball video using superimposed caption recognition, 2002. In Proc. of ACM International Conference on Multimedia.

[36] T. Zhang, H. Lu, and S. Li. Learning semantic scene models by object classification and trajectory clustering, 2009. In CVPR.

[37] X. Zhu. Semi-supervised learning literature survey, 2008. Computer Sciences Technical Report 1530, University of Wisconsin-Madison.

[38] D.S.B and M.P. Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences, 1980. IEEE Trans. ASSP.