Event Detection from Video Using Segment-Based Approach

PHAN LE SANG

Department of Informatics

School of Multidisciplinary Sciences

The Graduate University for Advanced Studies (SOKENDAI)

This dissertation is submitted for the degree of

Doctor of Philosophy

Tokyo, September 2015

for my loving parents . . .

Acknowledgements

This dissertation would have not finished without the support of many people.

I have been very fortunate to be supervised by Prof. Shin'ichi Satoh, who has a profound knowledge in my research area, who is always nice and provides whatever support for his students.

I am grateful to be co-supervised by Prof. Duy-Dinh Le, who always takes care of my research progress as well as other members in our group. I have been extremely lucky to be your student.

I also want to send my sincere gratitude to other professors in my Ph.D committee including Prof. Akihiro Sugimoto, Prof. Imari Sato and Prof. Gene Cheung. Without your valuable comments, my PhD would have taken longer to complete.

I would like to thank all of my friends who are always encouraging me to keep studying. Especially, I must thank my friends at National Institute of Informatics, Japan who have been my companies and/or my collaborators on the way pursuing my Ph.D.

Finally, I want to thank my parents and my love for your enduring support and love.

Abstract

Recognizing event in unconstrained videos is one of the most important tasks in multimedia retrieval. It has many potential applications such as video indexing, searching, and event recounting. However, this is a challenging task due to the large content variation and uncontrolled capturing condition. This leads to the fact that these videos often contain irrelevant information to the event of interest. The straightforward way to solve this problem is to decompose the original video into smaller segments and build the event detectors from these segment representations. This dissertation follows the aforementioned direction to study event detection methods in real videos. Essentially, we study three complementary approaches including *feature representation*, *feature aggregation* and *feature learning*.

In the first approach, we propose to use the segment-based (**SB**) *feature representation* to overcome the limitation of the traditional video-based approach. In the video-based approach, local features are extracted from the entire video and then aggregated to form the final video representation. However, this video-based representation is ineffective when used for realistic videos because the video length can be very different and the clues to determine an event may happen in only a small segment of the entire video. To handle this problem, our segment-based divides the original videos into segments for feature extraction and classification, while still keeping the evaluation at the video level. We investigate several strategies to divide a video into segments including non-overlapping uniform segment sampling, overlapping uniform segment sampling, and segments that based on the shot boundary detection. We also study the optimal segment length for event detection, which is close to the mean average length of the training videos.

The second approach handles the aforementioned problem by proposing a new video pooling strategy for *feature aggregation*. We consider a video as a layered structure where the lowest layer are frames, the top layer is the entire video, and the middle layers are the sequences of consecutive frames or the concatenation of lower layers. While it is easy to find local discriminative features in video from lower layers, it is non-trivial to aggregate these features into a discriminative video representation. In literature, people often use sum pooling to obtain reasonable recognition performance on artificial videos. However, the sum pooling technique does not work well on complex videos because the region of interests may reside within some middle layers. In this approach, we leverage the layered structure of video to propose a new video pooling method, named sum-max video pooling (**SM**), to handle this problem. Basically, we apply sum pooling at the low layer representation while using max pooling at the high layer representation. Sum pooling is used to keep sufficient relevant features at the low layer, while max pooling is used to retrieve the most relevant features at the high layer, therefore it can discard irrelevant features in the final video representation.

In the third approach, we focus on *feature learning* method to learn the key segments for video representation. In fact, a complex event can be recognized by observing necessary evidences. It is not easy to locate supportive evidences because they can happen anywhere in a video. A straightforward solution is to decompose the video into several segments and search for the evidences in each segment. This approach is based on the assumption that segment annotation can be assigned from its video label. However, this is a weak assumption because the importance of each segment is not considered. On the other hand, the importance of a segment to an event can be obtained by matching its detected concepts against the evidential description of that event. Leveraging this prior knowledge, we propose a new method, Event-driven Multiple Instance Learning (**EDMIL**), to learn the key evidences for event detection. We treat each segment as an instance and quantize the instance-event similarity into different levels of relatedness. Then the instance labels are learned by jointly optimizing the instance classifier and its related level. Finally the optimal instance classifiers are used to detect event.

We verify the effectiveness of our approaches on the large scale TRECVID Multimedia Event Detection 2010, 2011 and 2012 datasets. Our approaches can not only detect event, but also provide evidences for event detection. Compared to other segment-based approaches, our solutions achieve significant improvements. For example, when comparing in the MED 2011 dataset with a same setting, the baseline method (traditional video-based approach) has the average precision of 6.74%, while our methods (SB, SM and EDMIL) have the performance of 8.26%, 6.92% and 9.68% respectively.

Table of contents

List of figures

List of tables

Abbreviations

BOW Bag-of-Words

CNN Convolutional Neural Network

ESURF Extended Speeded Up Robust Features

FV Fisher Vector

GMM Gaussian Mixture Model

HOF Histogram of Optical Flow

HOG3G Histogram of 3D Gradients

HOG Histogram of Oriented Gradient

KLT Kanade–Lucas–Tomasi

MAP Mean Average Precision

SBD Shot Boundary Detection

MBH Motion Boundary Histogram

MED Multimedia Event Detection

MFCC Mel-frequency Cepstral Coeffcients

PCA Principal Component Analysis

SB Segment-based

SM Sum-Max Video Pooling

STIP Spatial-temporal Interest Points

SVM Support Vector Machines

TRECVID TREC Video Retrieval Evaluation

Chapter 1

Introduction

The moment you doubt whether you can fly, you cease for ever to be able to do it.

– J.M. Barrie, *Peter Pan*

1.1 Motivations

The evolution of internet has been changing our daily life. According to a report by the Internet World Stats [27], there is now more than 3 billions internet users, accounting for 40% of the world's population. The number of internet users are increasing rapidly and also keep producing a huge amount of internet data. It is important to analyze these data because it can provide valuable information about our daily activities. Among many interesting problems that need to be investigated, recognizing event in internet videos has been drawing a lot of attention in recent years [23, 59, 62, 83].

Recognizing event refers to the process of automatically identifying video clips that contain a particular event of interest. This is a challenging problem because we need to build computer system to recognize event not only from video metadata but also from its content. The detail definition of this task and its challenges will be described in Section 1.2 and Section 1.3 respectively.

Event recognition technologies are mainly employed in video retrieval systems to facilitate the retrieving progress. A video retrieval system that equipped such technologies can have numerous applications such as video search, video recommendation and video filtering. For example, below are some application scenarios:

- **Video search.** This is an important function in most of video sharing websites. Most of the time, these websites only provide a search interface that supports text queries. However, in order to do that, videos must have already been indexed based on its content and other metadata [98]. Using the provided interface, user can search for a specific tutorial such as "how to make a cake", "how to repair an appliance"; or some specific entertainment videos such as "a dog show" and "doing a magic trick".

- **Video recommendation.** It is also very important for video sharing websites to recommend videos that may appeal to the user. The longer the user stay on their websites, the higher the benefit. The recommendation is often based on the user's favorite videos or recently watched videos [15, 55]. From these input videos, the system will search for similar videos through their database within a short time. For example, when the user watch a video of "how to drive a car", they may also expected to watch similar events such as "how to park a car" and "common driving mistakes".

- **Video filtering.** In contrast to video recommendation, video filtering is also an important application of event detection technologies. There are certain event that the managers do not want them to be public, especially when a government want to establish a video censorship. For example, videos that teach "how to make a bomb" or "how to commit a suicide" should be removed from the retrieval results.

  Zillmann and Weaver [100] show that human tend to have violent responses when watching violent movies. In this case, event detection technology can be applied to filter out violent scenes in a movies. This technology has been employed in Facebook platform [63] to prevent the spread of a particular video over the internet, or to restrict the video from a particular type of audiences such as children, as shown in Fig. 1.1.

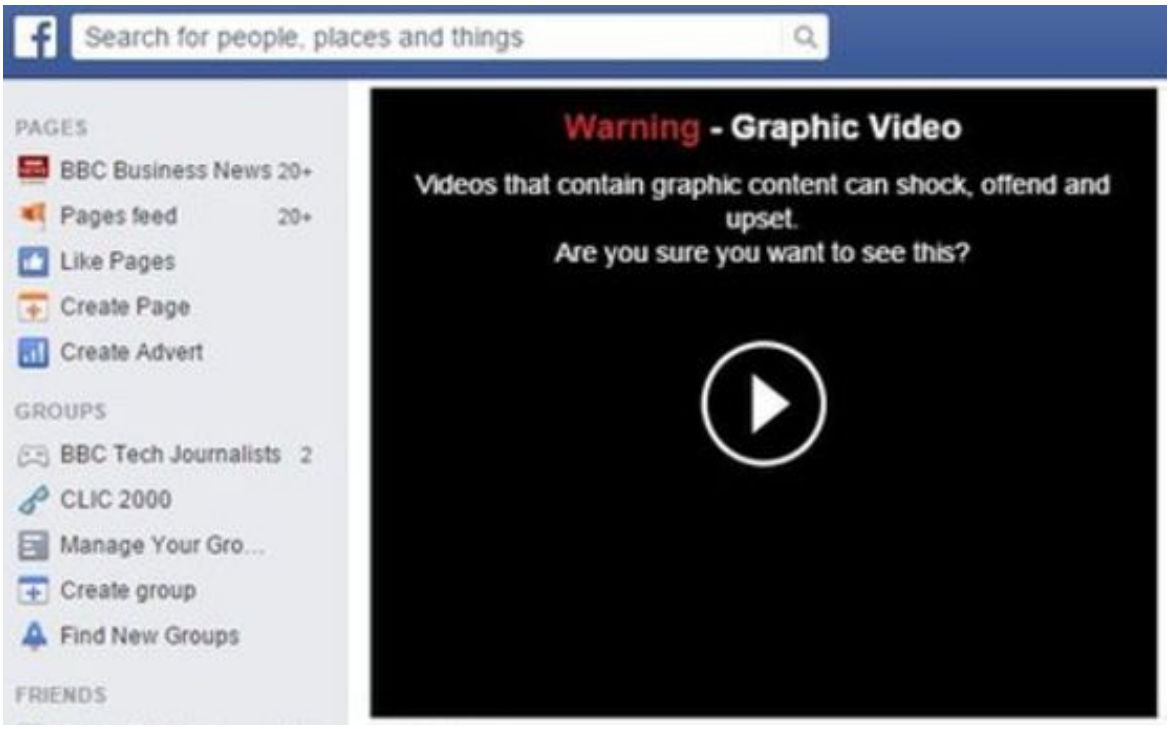


Fig. 1.1 Facebook has been placing warnings over violent videos posted to its site.

Motivated by these interesting applications, this dissertation aims to develop technologies for building an automatic event detection system. We will describe more about our research scope in the next section.

1.2 Problem Statement

This dissertation addresses the problem of recognizing complex event in videos. Basically, it is the process of automatically identifying video clips that contain a particular event of interest. There are two important characteristics of our target problem.

First, we are dealing with **complex event**. A complex event consists of various human activities and occurs in some particular settings. For example, "changing a vehicle tire" is an complex event that often happens at a garage or on street. This event contains several activities such as removing hubcap, turning lugwrench, unscrewing bolts and pulling rim out of tire. Complex event recognition differs from the traditional action recognition task in that it is the combination of multiple human actions or activities. It often contains various

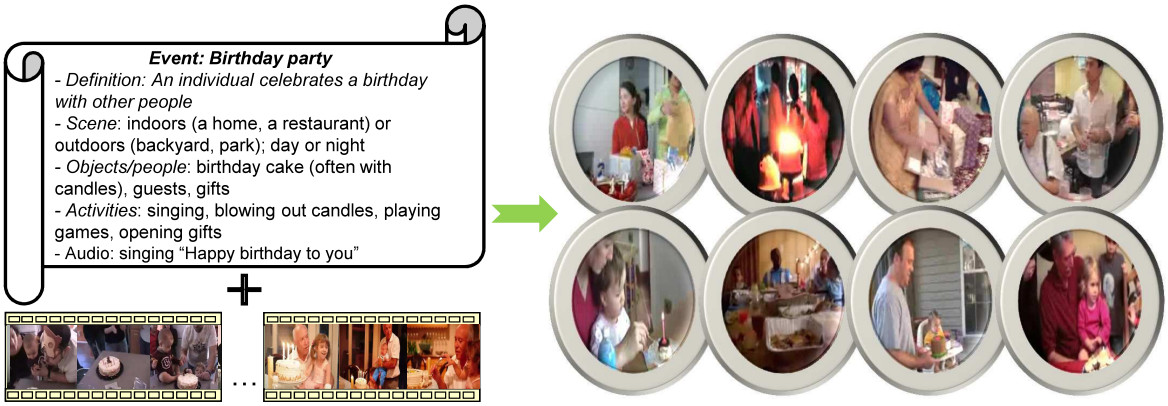


Fig. 1.2 Overview of an event detection from video system.

Fig. 1.3 Top: sequence of actions in the shoplifting event. Bottom: Examples of single action detection in KTH dataset.

interactions between human and objects in different scenes. Therefore, a complex event video is often longer than a single action video. Moreover, action videos are often captured in controlled environment, while complex event videos are often recorded by internet users, which is uncontrolled or arbitrary environment. Figure 1.3 shows the difference of complex event and single action detection. The top images are sequence of actions that happens in a shoplifting event, while the bottom images are examples of single action in the classic KTH dataset. Note that for the shoplifting event, the sequence of actions needed to be recorded in exactly the order from left to right. That means we not only deal with multiple actions, but also need to take into account the order that these actions happen.

Second, we are dealing with **multimedia data**. Internet videos can contain information from various mediums such as audio, visual and textual. Beside information from its content,

internet videos often come with user-provided metadata description such as titles, tags and descriptions. Traditionally, videos are indexed and retrieved based on this metadata information. However, text-based video retrieval systems face an intrinsic limitation that is the semantic gap between the content of the video and the information provided by the users. Moreover, this information is tend to be noisy and not always reliable. Therefore, we focus on utilizing multimedia data to build an effective event recognition system.

Due to the uncontrolled capturing condition of the complex videos, it is also interesting to know which parts of the video are important for recognizing event? How can we detect these parts? And suppose these parts do exist, how can we utilize them for complex event recognition? These challenging questions are also addressed in our dissertation.

1.3 Challenges

- **Large content variation.** The large content variation refers to the diversity of a complex event. Even though an event only involve with some specific objects, activities and scenes, the variety among within these classes is also very high. For example, "birthday party" is a complex event. This event can be happen during day or night and set in indoor (a home, a restaurant) or outdoor (a backyard, a park) environment. Typically, in a birthday party, the presence of a birthday cake is of the utmost importance. However, in the real world setting, even the birthday cake can be very different from video to video. Figure 1.4 shows some examples of birthday cake appear in internet videos.

  In terms of content variation, recognizing complex event is more challenging than other tasks such as instance search or copy detection . The instance search task aims to search for a certain specific person, object or location. These instances can have different views but it must belong to the same target of interest in the real world. The target of copy detection task is a little bit more flexible. It aims to detect a video segment that is derived from another video. The copy video can be derived from the original video by means of transformations such as addition, deletion and modification. To this end, the complex event detection task has the utmost content variation.

Fig. 1.4 The large variation of birthday cake in the birthday party event.

- **Uncontrolled capturing condition.** The uncontrolled capturing condition distinguish the complex event recognition task and the traditional action recognition task, which is often recorded in studio settings. As a result, techniques that work well for action recognition might no longer be effective for detecting complex event. For example, camera motion is one of the most frequent prominence in internet videos. Although popular motion features such as ESURF [91], STIP [42] and HOG3D [35] can effectively recognize action in studio videos, it shows limited performance in internet videos because it is not designed to handle camera motion. On the other hand, the Dense Trajectories feature proposed by Wang [90] takes into account the camera motion and demonstrates superior performance.

  One of the direct consequence of uncontrolled capturing condition is that user-generated videos often contain irrelevant information to the event of interest. In other words, different parts of the video have different levels of relatedness to a particular event. This leads to a challenging problem which is how to discard irrelevant information from the video representation. It is especially difficult when the annotation of each part of the video is almost not available. Figure 1.5 shows some examples of noisy information in internet videos.

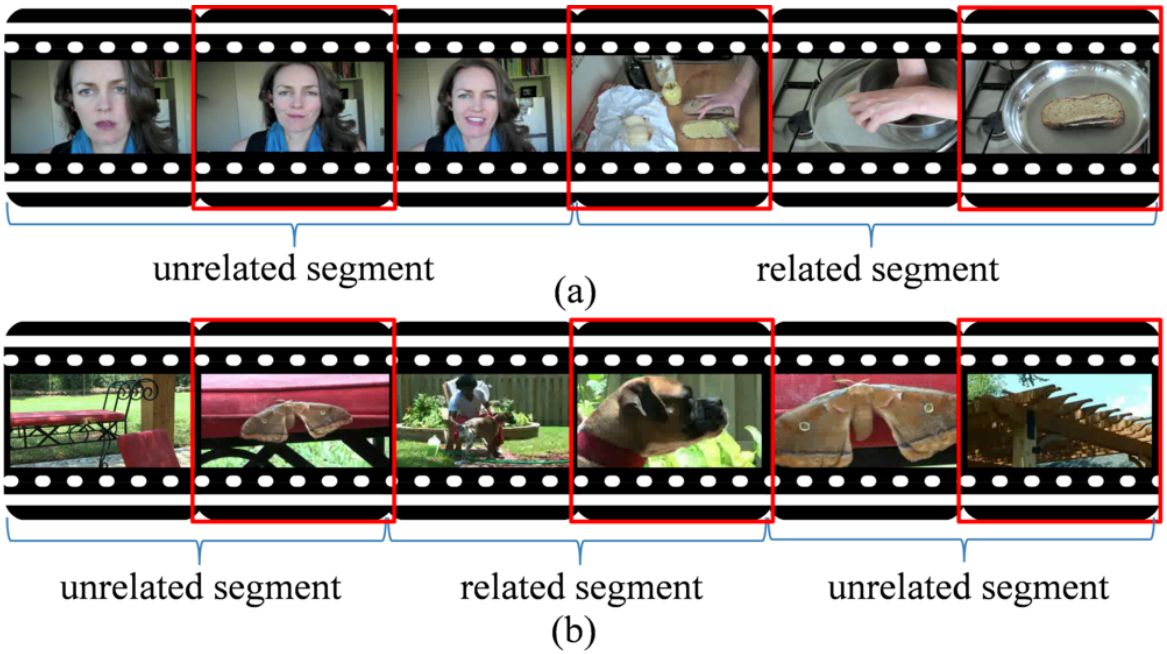


Fig. 1.5 (a) Example video for "making a sandwich" event: the related segment appears after a self-cam segment (unrelated); (b) example video for "grooming an animal" event: related segment is sandwiched between two unrelated segments. This kind of video is popular in realistic video datasets like MED. The frames with a red outlined box are examples of the extracted keyframes when using a keyframe-based approach, which suffers from both noise and missed extraction.

Fig. 1.6 Example of near-miss video for "Changing a vehicle tire" event. The first row shows some positive videos. The second row shows near-miss videos, which is very easy to be confused with positive ones, even for human.

- **Near-miss videos.** Near-miss video refers to a kind of video that is closely related to a particular event, however, it is not a positive instance of that event. Because a complex video is often composed by several objects or activities in some particular order, it might not be considered an event video if there is a lack of certain evidences. So a near-miss video can contain several evidences but not enough to define an event. This property of near-miss video often harm the performance of an event detection system. In fact, this kind of video is also prevalent in the setting of complex event recognition task. For example, "Changing a vehicle tire" is a complex event that involve one or more people to replace a tire on a vehicle. An event is not defined if the tire of the vehicle is not replaced. Examples of near-miss videos can be seen in Fig. 1.6.

- **Large scale video database.** Last but not least, we have to deal with big data as well. We have to accurately search for a particular event through a large video archive in a reasonable amount of time. In some complex event detection task such as TRECVID Multimedia Event Detection (MED) [72], the evaluation time is also limited, which forces the participants to care about the efficiency of their systems. In this contest, the participants need to prepare their system that is able to search for event on large collection of around 200,000 Internet videos, or 8,000 hours of videos [72].

1.4 Contributions

The main challenge that is addressed in this dissertation is "uncontrolled capturing condition". This challenge differentiates complex videos from artificial or studio setting videos. The straightforward approach to handle this challenge is to decompose the original video into small video segments and search for event evidences in these small segments. By following this research direction, we made three main contributions in our dissertation:

- We propose a new *feature representation* method, named segment-based representation (**SB**), to overcome the limitations of the traditional video-based approaches. The basic idea is to examine shorter segments instead of using the representative frames or entire video. We carry thorough experiments to verify our proposed method by investigating different strategies to decompose a video into segments. These strategies include uniform segment sampling and segments based on shot boundary detection.

- We propose a new *feature aggregation* method, called sum-max video pooling (**SM**), to deal with noisy information in complex videos. This pooling technique is based on the layer structure of video. Basically, we apply sum pooling at the low layer representation while using max pooling at the high layer representation. Sum pooling is used to keep sufficient relevant features at the low layer, while max pooling is used to retrieve the most relevant features at the high layer, therefore it can discard irrelevant features in the final video representation.

- We propose a new *feature learning* method, named Event-driven Multiple Instance Learning (**EDMIL**), to learn key evidences for complex event detection. We treat each segment as an instance and model it in a multiple instance learning framework [2], where each video is a “bag”. The instance-event similarity is quantized into different levels of relatedness. Intuitively, the most (ir)relevant instances should have higher (dis)similarities. Therefore, we propose to learn the instance labels by jointly optimizing the instance classifier and its related level.

1.5 Thesis Overview

The remaining of this dissertation is organized as follows:

Chapter 2 introduces some background that is related to our research. This background encompasses an introduction to TRECVID MED task and dataset. It also provide basic knowledge about some low level features and feature encoding methods, which is necessary to re-implement our system.

Chapter 3 presents our segment-based approach for complex event detection. At first we introduce the video-based approach and some of its limitation. After that we present the segment-based approach with several strategies to decompose a video into segments.

Chapter 4 presents our sum-max video pooling for complex event recognition. At first we introduce the layer structure of a video. Based on this layer structure, we propose a new video pooling technique which is a combination of sum pooling and max pooling.

Chapter 5 presents our method to detect event using the evidential description of an event. We also present a method to calculate the similarity between a video segment and an event based on textual description. This method can also provide evidences for event detection.

Chapter 6 concludes this dissertation by summarizing our contributions and discussing about the future work.

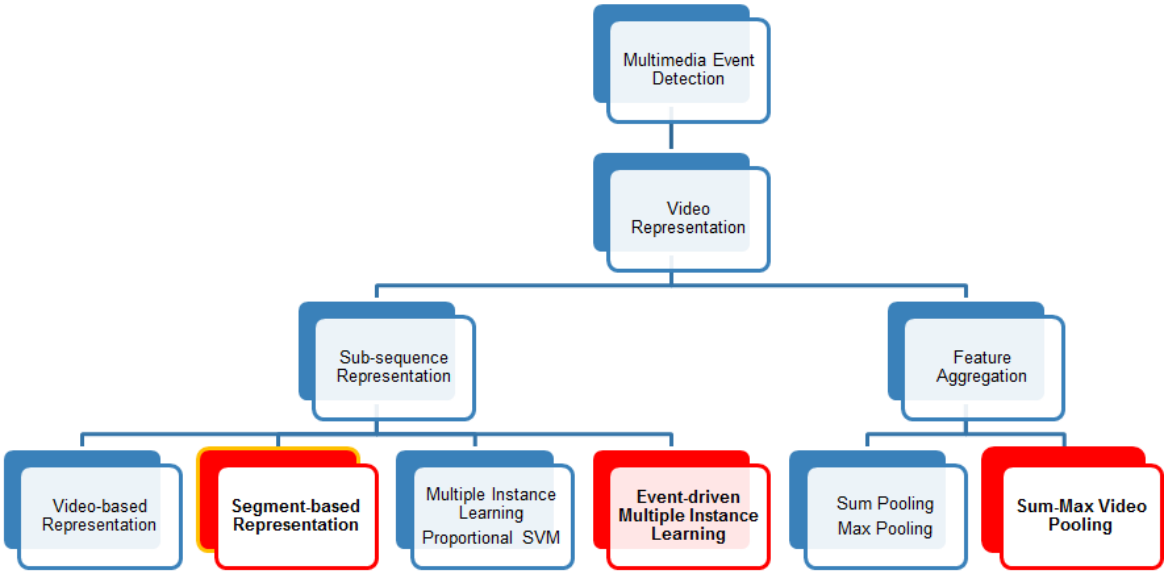


Fig. 1.7 Outline of our thesis. Our contributions are highlighted in the red boxes.

Chapter 2

Background

We can draw lessons from the past, but we cannot live in it.

– Lyndon B. Johnson

2.1 TRECVID Multimedia Event Detection

As introduced in Chapter 1, complex event recognition is an important computer vision research with many potential applications. In 2010 TRECVID community has proposed a new task, named "Multimedia Event Detection" [71] to advance the research and development in this area. The ultimate purpose of this task is to collect technologies for building a computer system that can quickly search for a particular event over a large video collection in a reasonable response time.

The task is defined as follows: "Given an event kit, find all clips that contain the event in a video collection" [71]. The event kit provides the event definitions along with some example videos of each event. At first, MED task defines an event:*is a complex activity occurring at a specific place and time; involves people interacting with other people and/or objects; consists of a number of human actions, processes, and activities that are loosely or tightly organized and that have significant temporal and semantic relationships to the overarching activity; and is directly observable*.

For a specific event of interest, a textual description is also provided to help developers generate the event search query. This textual description consists of following information: event name, event definition, event explication and evidential description. The event name is a mnemonic title of that event. The event definition provides a short definition of that event. Event explication is a long description which explains ambiguous terminologies in the event definition. Finally, the evidential description summarizes an event with its characteristics such as scene, object/people, activities and audio information. Table 2.1 shows textual description of an event in MED task.

2.2 Dataset

The TRECVID MED organizer also provides a standard benchmark for participants to evaluate their methods [71]. In the pilot task (MED 2010), there are only three events that are being tested[1]. These events are the following: (1) "*Assembling a shelter*": one or more people construct a temporary or semi-permanent shelter for humans that could provide protection from the elements. (2) "*Batting a run in*": within a single play during a baseballtype game, a batter hits a ball and one or more runners (possibly including the batter) scores a run. And (3) "*Making a cake*": One or more people make a cake. This collection is divided into two subsets including 1,744 videos for training and 1,724 videos for testing.

Since 2011, the number of test events has been increasing. New tested events as well as tested videos are added every year. For example, there are 5 training events (E001-E005) and 10 testing events (E006-E015) in MED 2011[2]. The number for MED 2012 is 20 testing events (E006-E015, E021-E030)[3]. These events are also kept in MED 2013 but more testing videos are added. In MED 2014, a different test set with 10 new events are introduced (E021-E040). List of all event names up to TRECVID MED 2014 can be found in Table 2.4. Since MED 2012, the evaluation set which contains around 98,000 test videos has been

[1] http://www.nist.gov/itl/iad/mig/med10.cfm
[2] http://www.nist.gov/itl/iad/mig/med11.cfm
[3] http://www.nist.gov/itl/iad/mig/med12.cfm

Table 2.1 Textual description for event “Attempting a board trick”

Event name	Attempting a board trick
Definition	One or more people attempt to do a trick on a skateboard, snowboard, surfboard, or other boardsport board.
Explication	Board sports are sports where a person stands, sits, or lays on a board and moves and controls the board. Tricks consist of intentional motions made with the board that are not simply slowing down/stopping the board or steering the board as it moves. Steering around obstacles or steering a board off of a jump and landing on the ground are not considered tricks in and of themselves. Common tricks involve actions like sliding the board along the top of an object (e.g. a swimming pool rim or railing), jumping from the ground or the surface of water into the air, and spinning or flipping in the air.
Evidential description	**scene**: outside, often in a skate park. **objects/people**: skateboard, snowboard, surfboard, ramps, rails, safety gear, crowds. **activities**: standing, sitting or laying on the board; jumping with the board; flipping the board and landing on it; spinning the board; sliding the board across various objects. **audio**: sounds of board hitting surface during trick; crowd cheering.

frozen. This collection is blind to all participants, which means they are not allowed to analyze these videos when tuning their systems.

Since MED 2014, the evaluation set has been doubled by adding around 100,000 test videos. An overview of all MED video collections is shown in Table 2.2. To the best of our knowledge, this is largest video dataset for event detection purpose. Because since MED 2012, the evaluation dataset has been frozen, most researchers conducts experiments on MED2010, MED2011 and MED2012 dataset [38, 39, 84, 86]. The detail information of these datasets can be seen in Table 2.3.

Table 2.2 Number of videos and video hours in the MED dataset up to 2014 [72].

Set		Number of video clips	Video duration (hours)
Development Data	RESEARCH	10,000	314
	10 Event Kits	1,400	74
	Transcription	1,500	45
Event Training Data	Event Background	5,000	146
	40 Event Kits	6,000	270
Test Data	MEDTest	27,000	849
	KindredTest	14,500	687
Evaluation Data	MED14Eval-Full	198,000	7,580
	MED14Eval-Sub	33,000	1,244
Total		244,000	9,911

Table 2.3 Detail information of MED2010, MED2011 and MED2012 dataset.

Dataset	No. Event	No. Train Videos	No. Test Videos	Total Videos	Total Hours
MED2010	3	1,744	1,724	3,468	110 hours
MED2011	10	1,331	31,822	33,153	1,100 hours
MED2012	25	3,878	1,938	5,816	250 hours

Table 2.4 List of event names in MED task from 2010-2014.

ID	Event name	ID	Event name
E001	Attempting a board trick	E021	Attempting a bike trick
E002	Feeding an animal	E022	Cleaning an appliance
E003	Landing a fish	E023	Dog show
E004	Wedding ceremony	E024	Giving directions to a location
E005	Working on a woodworking project	E025	Marriage proposal
E006	Birthday party	E026	Renovating a home
E007	Changing a vehicle tire	E027	Rock climbing
E008	Flash mob gathering	E028	Town hall meeting
E009	Getting a vehicle unstuck	E029	Winning a race without a vehicle
E010	Grooming an animal	E030	Working on a metal crafts project
E011	Making a sandwich	E031	Beekeeping
E012	Parade	E032	Wedding shower
E013	Parkour	E033	Non-motorized vehicle repair
E014	Repairing an appliance	E034	Fixing musical instrument
E015	Working on a sewing project	E035	Horse riding competition
E016	Doing homework or studying	E036	Felling a tree
E017	Hide and seek	E037	Parking a vehicle
E018	Hiking	E038	Playing fetch
E019	Installing flooring	E039	Tailgating
E020	Writing	E040	Tuning musical instrument

2.3 Feature for Event Detection

2.3.1 Image Features

Image features or still image features can be further classified into global and local features. Global features represent information of the whole frame (or image) while local features focus on some local invariant characteristics.

The most common global feature is color histogram, which is a representation of the distribution of colors in an image. Color histograms can potentially be identical for two images with different object content which happens to share color information. Another popular global feature is GIST [66]. GIST descriptor describes the dominant spatial structure of a scene in a low dimensional representation.

For local image features, Scale-Invariant Feature Transform (SIFT) has become a standard feature for many image classification tasks. It is proposed by Lowe in [50] to find local maximum of Difference of Gaussians (an approximation of Laplacian of Gaussian) in space and scale. It is a scale invariant local feature, simple and efficient. Other variants of SIFT take into account the keypoint extraction methods, such as Hessian-Laplace interest points detector [56] and dense sampling [64]. In both strategies, local features are extracted from multiple scales by using the Gaussian scale space [56]. In the case of dense sampling, the key points are densely sampled on a grid with a step size of 6 pixels. Once a key point is detected, it is described using the standard SIFT [50]. It is also acknowledged that other descriptors such as RGB-SIFT, Opponent-SIFT, and C-SIFT [7] can be complementary with the standard SIFT descriptor [96].

2.3.2 Motion Features

Motion features have been widely developed for various action recognition tasks. Because event video may contain multiple actions, it is reasonable to employ motion information for event detection. Depend on the extraction methods, motion features can be classified into two categories: methods based on interest points and methods based on tracking.

Methods based on interest points. The Harris3D detector was proposed by Laptev and Linderberg [42]. This is an extension of Harris2D detector which is used to detect corners in image. Generally the proposed detector will detect points which have significant change in both spatial and temporal direction. First, the spatial temporal second-moment matrix averaged using a Gaussian weighting function g at spatial scale σ_i and temporal scale τ_i is defined as below:

$$\mu = g(.;\sigma_i^2,\tau_i^2,) * \begin{pmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_t & L_yL_t & L_t^2 \end{pmatrix} \tag{2.1}$$

Second, a new response function is proposed to measure the motion in 3D:

$$H = \det(\mu) + k\,\mathrm{trace}^3(\mu), \tag{2.2}$$

Finally, interest points are those which maximize the response function H. To do this, the authors use a corresponding eigenvalue problem. The response function H can be rewrote as below:

$$H = \lambda_1\lambda_2\lambda_3 - k(\lambda_1+\lambda_2+\lambda_3)^3 \tag{2.3}$$

Interest points are those have large eigenvalue $\lambda_1,\lambda_2,\lambda_3$ which means large variation in both spatial and temporal domain.

The Cuboid detector is based on temporal Gabor filters and was proposed by Dollár et al. in [17]. The response function has the form:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{2.4}$$

where g(x,y;σ) is the 2D spatial Gaussian smoothing kernel, and h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters which are applied temporally. The Gabor filters are defined by:

$$h_{ev}(t;\tau,\omega) = -cos(2\pi t\omega)e^{\frac{-t^2}{\tau^2}} \tag{2.5}$$

$$h_{od}(t;\tau,\omega) = -sin(2\pi t\omega)e^{\frac{-t^2}{\tau^2}} \tag{2.6}$$

with ω=4/τ. The two parameters σ and τ of the response function R correspond roughly to the spatial and temporal scale of the detector. Interest points are the local maxima of the response function R.

The Hessian detector was proposed by Willems et al. [91] as a spatio-temporal extension of the Hessian saliency measure used for blob detection in images. The detector measures the saliency with the determinant of the 3D Hessian matrix. The position and scale of the interest points are simultaneously localized without any iterative procedure. In order to speed up the detector, the authors used approximative box-filter operations on an integral video structure. Each octave is divided into 5 scales, with a ratio between subsequent scales in the range 1.2 – 1.5 for the inner 3 scales. The determinant of the Hessian is computed over several octaves of both the spatial and temporal scales. A non-maximum suppression algorithm selects joint extrema over space, time and scales: (x, y, t, σ, τ).

Methods based on tracking. Methods based on tracking process the video frame by frame. Trajectories are often extracted after tracking for fixed number of frame length. Different features may different in both the extraction and description method. There are several methods to extract trajectories. In [53], a standard Kanade–Lucas–Tomasi (KLT) tracker is used to track features (using "good features to track") over a video. A fixed number of features (typically 100) is initialized for tracking. Features are replaced as necessary when tracks are lost. The output of this tracking is a trace of (x; y) pairs for each feature.

In [21], trajectories are generated by tracking dense SIFT points frame by frame. The reason for using dense sampling, according to the author, is to provide sufficient points to group similar motions into meaningful body parts. First, points are sampled on a regular grid spacing with 5 pixels, then each image patch is represented by a SIFT descriptor. The correspondence between key-points in successive frames is established by nearest neighbor distance ratio matching.

Trajectories can also be tracked using dense optical flow. This method proposed by Wang in [89, 90]. The trajectories are obtained by tracking densely sampled points using optical

flow fields. First, feature points are sampled on a grid spaced by 5 pixels and at multiple scales spaced by a factor of $1/\sqrt{2}$. Then features are tracked in each scale separately. Each point $P_t = (x_t, y_t)$ at frame t is tracked to the next frame $t+1$ by median filtering in a dense optical flow field $\omega = (u_t, v_t)$.

Tracking using KLT can be difficult for initialization step, particularly when the scene contains distracting objects. Moreover, sparse tracking methods like KLT might not capture enough information of a moving object. That is why the dense trajectory approach [89] yields better result than sparse trajectory extraction approach. However, tracking object is still challenging in real world environment, where occlusion between moving objects are popular. In that case, the tracker output tends to be noisy. On another hand, matching dense SIFT descriptors is computationally very expensive [48] and, thus, infeasible for large video datasets.

Trajectory descriptors. The trajectory descriptors are computed within a space-time volume around the trajectory (see Fig. 2.1). The size of the volume is NxN pixels and L frames. To embed structure information in the representation, the volume is subdivided into a spatio-temporal grid of size n_σ x n_σ x n_τ. The default parameters for our experiments are N = 32, $n_\sigma = 2$, $n_\tau = 3$.

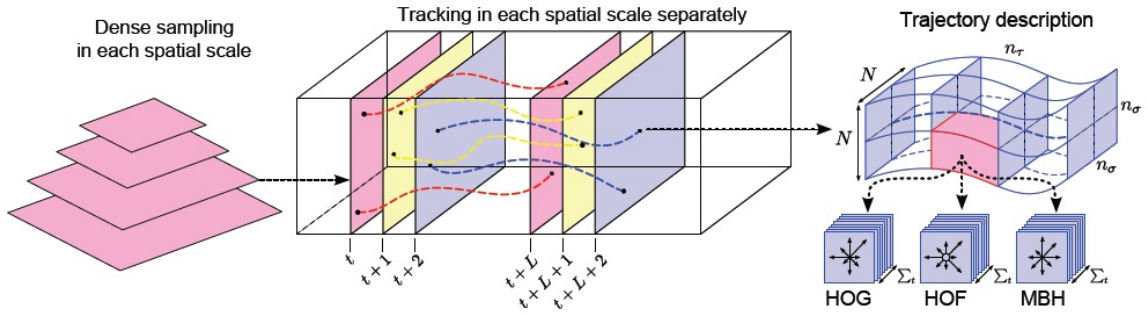


Fig. 2.1 Illustration of dense trajectory description. Left: Feature points are sampled densely for multiple spatial scales. Middle: Tracking is performed in the corresponding spatial scale over L frames. Right: Trajectory descriptors are based on its shape represented by relative point coordinates as well as appearance and motion information over a local neighborhood of N x N pixels along the trajectory. In order to capture the structure information, the trajectory neighborhood is divided into a spatio-temporal grid of size n_σ x n_σ x n_τ [89].

The HOGHOF [42] descriptor has shown excellent results on a variety of datasets. HOG (histograms of oriented gradients) focuses on static appearance information, whereas HOF (histograms of optical flow) captures the local motion information. HOGHOF descriptors are computed along the dense trajectories (see Fig. 2.1). For both HOG and HOF, orientations are quantized into 8 bins using full orientations, with an additional zero bin for HOF (i.e., in total 9 bins). Both descriptors are normalized with their L_2 norm.

The MBH descriptor is proposed by Dalal et al. [14] for human detection, where derivatives are computed separately for the horizontal and vertical components of the optical flow. This descriptor encodes the relative motion between pixels (See Fig. 2.2).

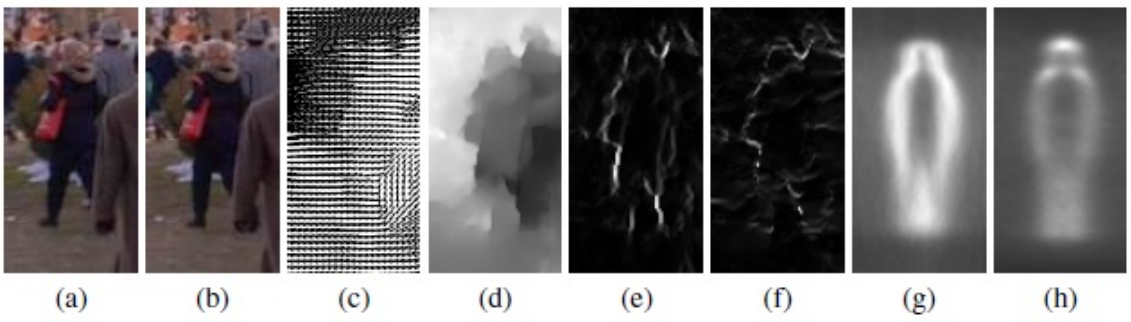


Fig. 2.2 Illustration of the MBH descriptor. (a,b) Reference images at time t and t+1. (c,d) Computed optical flow, and flow magnitude showing motion boundaries. (e,f) Gradient magnitude of flow field I_x, I_y for image pair (a,b). (g,h) Average MBH descriptor over all training images for flow field I_x, I_y [14].

The MBH descriptor separates the optical flow field $I_\omega = (I_x, I_y)$ into its x and y component. Spatial derivatives are computed for each of them and orientation information is quantized into histograms, similarly to the HOG descriptor (have 8-bin histogram for each component). Finally, these two histograms are normalized separately with the L_2 norm. Since MBH represents the gradient of the optical flow, constant motion information is suppressed and only information about changes in the flow field (i.e., motion boundaries) is kept. This is a simple way to eliminate noise due to background motion. This descriptor yields excellent results when combined with dense trajectory features.

As shown by Wang et al. [89, 90], the dense trajectory feature is one of the best for action classification. In particular, it is an efficient way to remove camera motion. Violent

scenes of Hollywood movies tend to have a lot of action and different effects. We use the dense trajectory feature to capture this information. Trajectories are obtained by tracking densely sampled points in the optical flow fields. As suggested by Wang [89, 90], we use Histogram of Oriented Gradient (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH) to describe each trajectory. HOG captures the appearance of a moving object, whereas HOF captures its speed. The last descriptor, MBH, captures the boundaries of motion and is good for handling camera motion.

2.3.3 Audio Features

We use the popular Mel-frequency Cepstral Coeffcients (MFCC) [75] for extracting audio features. We set the window to 25 ms and the step size to 10 ms. 13-dimensional MFCC vectors along with their first and second derivatives are used for representing each audio segment. Raw MFCC features are also encoded using BoW. Note that this configuration was used by the winning teams (AXES/LEAR) of the TRECVID Multimedia Event Detection 2013 [1] and THUMOS Challenge 2014 [70].

We investigated several ways to extract MFCC features from audio channel. These MFCC libraries are used in our evaluation: VoiceBox audio toolkit [6], Yaafe audio library [52] and the RASTA-PLP library [20]. We found that the RASTA-PLP implementation achieved slightly better performance than others. Moreover, we did not observe significant improvement when changing parameters such as window length and step between successive windows. So we kept using the default setting in the RASTA-PLP implementation.

2.3.4 Deep Learning Features

Deep learning has been drawing a lot of attention after the seminal work of Krizhevsky et al. [37]. They proposed a deep learning framework that significantly outperforms previous state-of-the-art methods on the ImageNet benchmark [16]. This is a variant of multilayer perceptrons (MLP) [77], where a larger number of layers can be incorporated into the network (Fig. 2.3). The success of deep learning is due to the explosion of big data, Convolutional

Neural Network (CNN) [45, 46], as well as major improvements for training the network [4, 26].

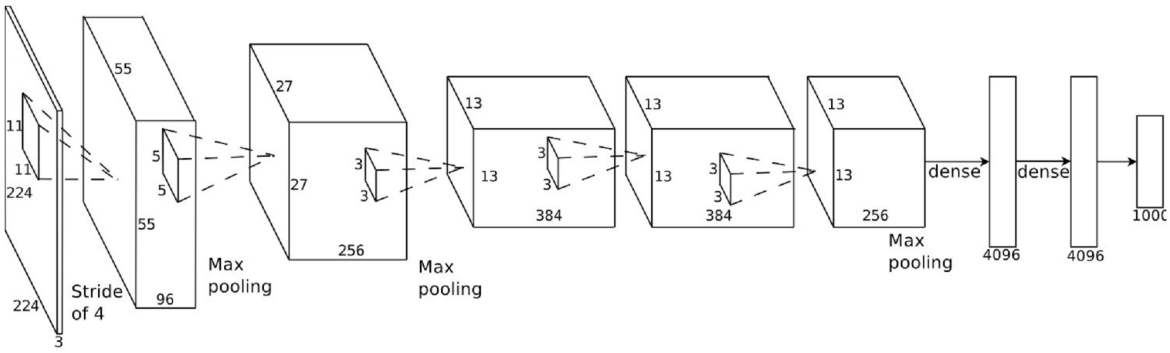


Fig. 2.3 Illustration of deep learning architecture that was used in [37].

There are basically three ways to apply deep learning for event detection from video. (1) Using pre-trained deep models to extract video features [85]. Note that if the model was trained on image collection [9, 37, 80, 99], it can be used to extract image features for each sample video frames. Video level features can be obtained by aggregating from all of its keyframe-based featrues. (2) The second approach is to train a deep learning model directly on event video collection. As a result, we can have an end-to-end network for the event detection. However, deep learning techniques for video is still not matured, and training a deep model with a limited number of positive labels might be not effective. (3) The third method is fine-tuning for event detection data on top a pre-trained model. This is the most common approach when applying deep learning to a new application. However, this approach requires the pre-trained model should be trained on a dataset that is similar to video event collection, otherwise, deep learning might not be applicable.

In this dissertation, we used the popular DeepCaffe [29] framework to extract keyframe features. We used the pre-trained deep model provided by DeepCaffe. This model was trained on ImageNet 1,000 concepts [16]. The protocol for training it is described in [29]. As suggested in [37], we selected the last three fully connected layers for the feature representation. The third and second-to-last layers have 4,096 dimensions, while the last layer has 1,000 dimensions corresponding to the 1,000 concept categories in the ImageNet dataset.

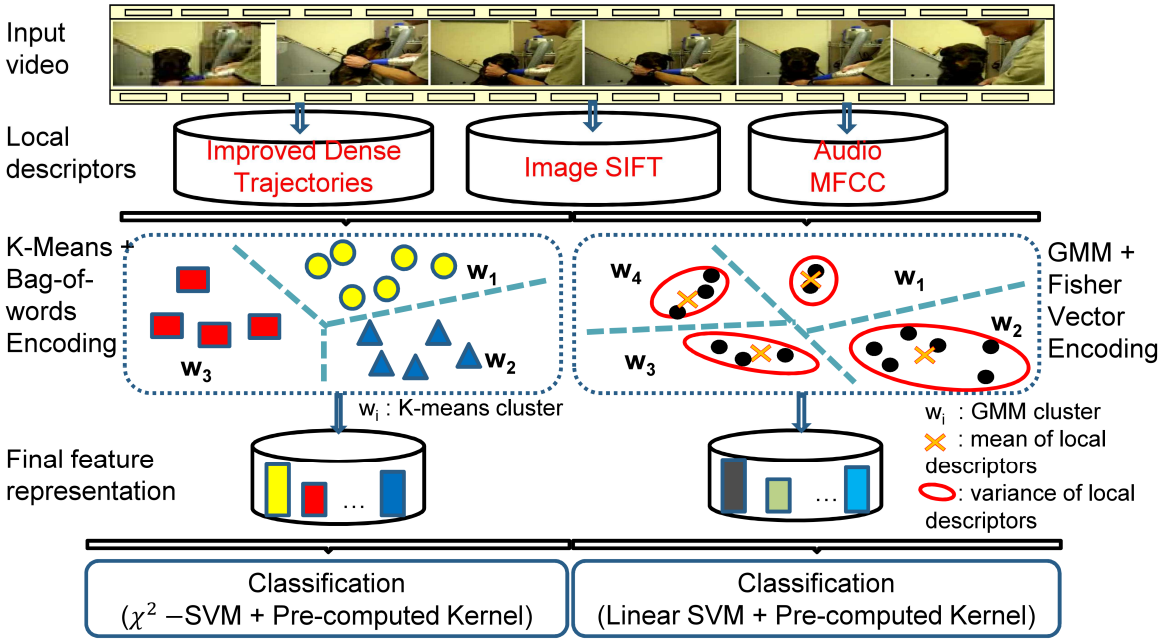


Fig. 2.4 General MED framework

2.4 General Framework

We design a unified framework to evaluate the performance of individual features and their combination (see Figure 2.4). We made it flexible in that we can easily test different features. We also designed it in components, i.e., pre-processing, feature extraction, feature encoding, feature classification, and feature fusion, so that each component could be evaluated separately while keeping the others intact. In particular, it consists of the following components.

2.4.1 Pre-Processing

The pre-processing component prepares the data for the processing in the other components. First, the video is resized to a width of 320 pixels, and its height is scaled so that the aspect ratio is kept the same. All features are extracted from the resized video.

To get the image features, keyframes are sampled from the shots at one frames every two seconds. This rate seems to be a good tradeoff between time and accuracy (as suggested in

[33]). Blank keyframes, i.e., ones filled with a single color, are removed because they do not contain informative features.

To get the audio features, the audio channel is extracted from the original video and saved as a file in the standard WAV format. The audio features can be extracted from this file.

2.4.2 Feature Extraction

The feature extraction component aims to make a discriminative vector representation for each shot extracted in the pre-processing step. The extraction method depends on what type of feature will be used. To conduct comprehensive evaluations of features for VSD, our framework supports a large variety of features, including global and local visual features. Global features capture the global statistics of each extracted shot. These statistics can be calculated directly from sub regions of a sampled frame and concatenated to form the vector representation for that frame, before being aggregated into the final representation for each shot. It is more complicated to calculate the feature vector representation for local features. The number of local features varies from frame to frame; therefore, it requires a special encoding technique, which will be described in Section 2.4.3.

Besides global and local features, our evaluation framework supports a number of other features. In particular, audio features can be extracted from pre-defined temporal windows. The features of each window provide local audio characteristics at that temporal location. This means audio features can be considered as local. Another kind of feature is a mid-level feature made using concept detectors. We use general concepts taken from off-the-shelf datasets [16]. In addition, our framework supports state-of-the-art deep learning features, which are extracted from a pre-trained model. A description of each feature is presented in Section 2.3.

2.4.3 Feature Encoding

Bag-of-word model

As for local features, we use the popular Bag-of-Words (BOW) model to generate a fixed-length representation from local descriptors. This model was initially used to represent text documents [24], and it was first used to represent images by Csurka et al. [12]. Its extension to motion and audio features is straightforward [34, 81].

We used the experiment setup described in [32] to make our bag-of-words models. We set the codebook size to 1,000, because in [32], performance did not significantly improve when the larger codebooks were used, and a smaller codebook can significantly reduce the computational time for feature encoding as well as feature learning. In order to train the codebook, we randomly selected 1 million local descriptors and clustered them using the K-means algorithm. The local descriptors were assigned to each codeword in a soft-weighting manner [31] to improve the discriminative power of the encoded feature.

The main drawback of the bag-of-words model is that it does not incorporate spatial information. The simplest way to overcome this problem is to partition the image into sub-regions and encode local features in each region independently. After that, features from all regions are concatenated into a single feature vector. There are many ways to partition an image into sub-regions. To this end, we follow [32] and [44] and use 2 x 2 and 1 x 3 spatial configurations. We found that these spatial configurations are good trade-offs between performance and computational cost of the high-dimensional feature vector.

Fisher Vector Encoding

The Fisher vector (FV) was first used for image classification in [28]. It has since been used for action recognition, such as in [83] and [90]. Fisher vector encoding can be considered to be an extension of Bag-of-words encoding. Unlike a bag of features, the Fisher vector encodes both first- and second-order statistics between the local descriptors and the codebook. As a result, it is much longer than the BoW feature when using the same codebook.

Different from bag-of-words encoding, which often uses k-means to train the codebook, the Fisher vector often uses the Gaussian Mixture Model (GMM) to encode the relative position of each local descriptor to each mixture center. The relatively large expressiveness of the Fisher vector means it can achieve comparable performance to that of BoW while using a much smaller codebook [78, 83].

In our experiment, we set the number of Gaussians in the GMM model to K = 256. Then we randomly selected 1,000,000 local descriptors for training the model. As suggested in [73], it is better to reduce the local feature dimension by using Principal Component Analysis (PCA). The normalization of the output feature is also very important. Following the recommendation in [73], we applied power normalization with $\alpha = 0.5$ followed by L2-normalization to the Fisher vector.

2.4.4 Learning

Support Vector Machine (SVM) is a standard machine learning algorithm for image and action recognition tasks. Therefore, we also use it in our experiments. To this end, we use the LibSVM [8] for training and testing. Because we often deal with event collection with more than two events, we simply adopt the one-vs.-rest scheme to solve the multi-class classification problem.

For features encoded using the "bag-of-words" model, we use the χ^2 kernel to calculate the distance matrix. The optimal (C;g) parameters for learning Support Vector Machine (SVM) classifiers are found by conducting a grid search with five-fold cross validation on the original dataset. For features that are encoded with the Fisher vector, we use LibSVM with linear kernel. In this case, we perform a five-fold cross-validation to obtain the learning parameter C.

2.4.5 Fusion Scheme

Fusing information from different media seems to be a natural way to handle multimedia content. Fusing multi-modal information has been used for multimedia event detection in

recent works such as [40, 59, 62, 65]. The different types of multimedia data have their own characteristics, so it is also natural that they would have different fusion strategies [94]. Here, we chose to use late fusion with an average weighting scheme for all features [82]. This simple fusion strategy has demonstrated stable performance across many event detection collections as well as different set of features.

Chapter 3

Event Detection Using Segment-based Feature Representation

Concentrate all your thoughts upon the work at hand. The sun's rays do not burn until brought to a focus.

– Alexander Graham Bell

3.1 Introduction

Multimedia Event Detection (MED) is a challenging task in TREC Video Retrieval Evaluation (TRECVID)[1]. The task is defined as follow: given a collection of test videos and a list of test events, indicate whether each of the test events is present in each of the test videos. The aim of MED is to develop systems that can automatically find video containing any event of interest, assuming only a limited number of training exemplars are given.

The need for such MED systems is rising because a massive number of videos are produced every day. For example, more than 3 million hours of video are uploaded and over 3 billion hours of video are watched each month on YouTube[2], the most popular video

[1] http://trecvid.nist.gov/

[2] http://www.youtube.com/t/press_statistics

sharing website. What is needed are the tools for automatically processing the video content and looking for the presence of a complex event in such unconstrained capturing videos. Automatic detection of complex events has great potential for many applications in the field of web video indexing and retrieval. In practice, a viewer may only want to watch goal scenes in a long football video, a housewife may need to search for videos that teach her how to make a cake, a handyman may look for how to repair an appliance, or a TV program manager may want to remove violent scenes in a film before it is aired.

However, detecting events in multimedia videos is a difficult task due to both the large content variation and uncontrolled capturing conditions. The video content is extremely diverse even in a same event class. The genres of video are also very varied, such as interviews, home videos, and tutorials. Moreover, the number of events is expected to be extensive for large scale processing. Each event, in its turn, can involve a number of objects and actions in a particular setting (indoors, outdoors, etc). Furthermore, multimedia videos are typically recorded under uncontrolled conditions such as different lighting, viewpoints, occlusions, complicated camera motions and cinematic effects. Therefore, it is very hard to model and detect of multimedia events.

The most straightforward approach toward building a large scale event detection system is using a bag-of-words (BoW) model [12]. There are two types of BoW representations that are used for MED: BoW representation at the keyframe level and BoW representation at the video level. The first method is employed for still image features where the keyframes are often extracted at a fixed interval. The second method is employed for motion features where moving patterns from the entire video are extracted. These methods are respectively referred to as keyframe-based [25, 33, 54] and video-based [25, 33] in this chapter. Although these methods can obtain reasonable results, they all suffer from severe limitations. For the keyframe-based approach, temporal information is not incorporated in the model. Moreover, it is possible that important keyframes are missed extraction. Extracting more keyframes can tackle this problem but the scalability is also a problem for concern. On the other hand, the video-based approach is most likely to suffer from noise. We found that the video length is very different from video to video (even from videos of the same event class). In addition,

the clues to determine an event may appear within a small segment of the entire video. Thus, comparing the BoW representation of two videos is unreliable because it may contain unrelated information. Figure 1.5 illustrates these limitations for both approaches.

In this chapter, we propose using a segment-based approach to overcome the limitations of both the keyframe-based and video-based approaches. The basic idea is to examine shorter segments instead of using the representative frames or entire video. We can reduce the amount of unrelated information in the final representation, while still benefiting from the temporal information by dividing a video into segments. In particular, we investigate two methods to cut a video into segments. The first method is called uniform sampling, where every segment has an equal length. We choose different segment lengths and use two types of sampling: non-overlapping and overlapping. The overlapped configuration is used to test the influence of dense segment sampling. The second method divides the video based on the shot boundary detection to take into account the boundary information of each segment. Once segments are extracted, we use dense trajectories, a state-of-the-art motion feature proposed by Wang [89], for the feature extraction. After that, a BoW model is employed for the feature representation. The experimental results on TRECVID MED 2010 and TRECVID MED 2011 showed the improvement of the segment-based approach over the video-based approach. Moreover, a better performance can be obtained by using the overlapping sampling strategy.

The rest of this chapter is organized as follows. Section 3.2 introduces the related work. Section 3.3 gives an overview of the dense trajectory motion feature and our segment-based approach. The experimental setup including an introduction to the benchmark dataset and the evaluation method are presented in Section 3.4. Then, in Section 3.5, we present and analyze our experimental results. Detailed discussions of these results are presented in Section 3.6. Finally, Section 3.7 concludes this work with discussions on our future work.

3.2 Related Work

Challenges began from TRECVID 2010[3], and Multimedia Event Detection has drawn the attention of many researchers. Seven teams participated in the debut challenge and 19 teams participated the following year (MED 2011). Many MED systems have been built and different strategies have been used for the event detection system.

Columbia University (CU) team achieved the best result in TRECVID MED 2010. Their success greatly influenced later MED systems. In their paper [33], they answered two important questions. The first question was, "What kind of feature is more effective for multimedia event detection?". The second one was, "Are features from different feature modalities (e.g., audio and visual) complementary for event detection?". Different kinds of features have been studied, such as SIFT [50] for the image feature, STIP [41] for the motion feature and MFCC (Mel-frequency cepstral coefficients [47]) for the audio feature to answer the first question. In general, the STIP motion feature is the best single feature for MED. However, the system should combine strong complementary features from multiple modalities (both visual and audio) in order to achieve better results.

The IBM team [25] achieved the runner-up MED system in TRECVID 2010. They incorporated information from a wide range of static and dynamic visual features to build their baseline detection system. For the static features, they used the local SIFT [50], GIST [67] descriptors and various global features such as Color Histogram, Color Correlogram, Color Moments, Wavelet Texture, etc. They used the STIP [41] feature with a combined HOG-HOF [43] descriptor for the dynamic feature.

The Nikon MED 2010 system [54] is also a remarkable system due to its simple but effective solution. They built a MED system based on the assumption that a small number of images in a given video contain enough information for event detection. Thus, they reduced the event detection task to the classification problem for a set of images, called keyframes. However, keyframe extraction is based on a scene cut detection technique [22] that is less reliable in realistic videos. Moreover, the scene length is not consistent, which may affect the detection performance.

[3] www.nist.gov/itl/iad/mig/med10.cfm

The BBN Viser system [60] achieved the best performance at TRECVID MED 2011. Their success confirmed the effectiveness of the multiple modalities approach for multimedia event detection. In their work, they further investigated the performance of the appearance features (e.g., SIFT [50]), color feature (e.g. RGB-SIFT [87]), and motion (e.g., STIP [41]), and also MFCC [47] based audio features. Different kinds of fusion strategies have been explored, from which the novel non-parametric fusion strategy based on a video specific weighted average fusion has shown promising results.

In general, most systems used the multiple modalities approach to exploit different visual cues to build their baseline detection systems. Static image characteristics are extracted from frames within provided videos. Colombia University's results [33] suggest that methods for exploiting semantic content from web images, such as [18] and [33], are not effective for multimedia event detection. For motion characteristics, most systems employed the popular STIP proposed by Laptev in [41] for detecting complex actions. Other systems also took into account the HOG3D [35] and MoSIFT [11] motion features. All these systems used a video-based approach for the motion features, i.e., the motion features are extracted from the entire video. IBM's MED system [25] also applied the video-based approach but the video was downsampled to five frames per second. One drawback of this video-based approach is that it may encode unrelated information in the final video representation. In a long video, the event information may happen during a small segment, and the information from the other segments tends to be noisy. That is why it is important to localize the event segment (i.e., where the event happens). This problem has been thoroughly investigated by Yuan et. al. [97]. Yuan proposed using a spatio-temporal branch-and-bound search to quickly localize the volume where an action might happen. In [93], Xu proposed a method to find optimal frame alignment in the temporal dimension to recognize events in broadcast news. In [19], a transfer learning method is proposed to recognize simple action events. However, these works are not applicable for complex actions in multimedia event videos.

Different from other approaches, we use a segment-based approach for the event detection. We did not try to localize the event volume like Yuan in [97]. In a simpler way, we use a uniform sampling with different segment lengths for our evaluation. We also investigate

the benefit of using the shot boundary detection technique in [22] for dividing video into segments. Moreover, an overlapped segment sampling strategy is also considered for a denser sampling. To the best of our knowledge, no MED system has previously used this approach. We evaluate its performance using the dense trajectories motion feature that was recently proposed by Wang in [89]. The dense trajectories feature has achieved state-of-the-art performances for various video datasets, including challenging datasets like Youtube Action [49] and UCF Sports [76]. In TRECVID MED 2012, the dense trajectories feature was also widely used by top performance systems such as AXES [68], and BBNVISER [61]. We use the popular "bag-of-words" model in [12] as our feature representation technique. Finally, we use a Support Vector Machine (SVM) classifier for the training and testing steps.

3.3 Dense Trajectories and Segment-based Approach

We introduce the dense trajectory motion feature proposed by Wang in [89] in this section. We additionally briefly review the trajectory extraction and description method. A detailed calculation of all the related feature descriptors, especially for Motion Boundary Histogram, is also presented. Our segment-based approach for motion features is introduced at the end of this section.

3.3.1 Dense Trajectories

Trajectories are obtained by tracking the densely sampled points using the optical flow fields. First, the feature points are sampled on a grid with a step size of 5 pixels and at multiple scales spaced by a factor of $1/\sqrt{2}$. Then, the feature points are separately tracked in each scale. Each point $P_t = (x_t, y_t)$ at frame t is tracked to the next frame $t+1$ by using median filtering in a dense optical flow field $\omega = (u_t, v_t)$:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega)|_{(\bar{x}_t, \bar{y}_t)}, \tag{3.1}$$

where M is the median filter, and $(\bar{x}_t, \bar{y}_t)$ is the rounded position of (x_t, y_t).

After extracting a trajectory, two kinds of feature descriptors are adopted: a trajectory shape descriptor and a trajectory-aligned descriptor.

Trajectory shape descriptor: The trajectory shape descriptor is the simplest one for representing an extracted trajectory. It is defined based on the displacement vectors. Given a trajectory of length L, its shape is described by the sequence S = (ΔP_t ,..., ΔP_{t+L-1}), where $\Delta P_t = P_{t+1} - P_t = (x_{t+1} - x_t, y_{t+1} - y_t)$. The resulting vector is then normalized by the sum of the magnitudes of the displacement vectors:

$$S' = \frac{(\Delta P_t, ..., \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\Delta P_j\|} \tag{3.2}$$

Trajectory-aligned descriptor: More complex descriptors can be computed within a space-time volume around the trajectory. The size of the volume is NxN spatial pixels and L temporal frames. This volume is further divided into a n_σ x n_σ x n_τ grid to encode the spatial-temporal information between the features. The default settings for these parameters are N = 32 pixels, L = 15 frames, n_σ = 2, and n_τ = 3. The features are separately calculated and aggregated in each region. Finally, the features in all regions are concatenated to form a single representation for the trajectory. Three kinds of descriptors have been employed for representing trajectory following this design: The Histogram of Oriented Gradient (HOG), which was proposed by Dalal et al. in [13] for object detection, The Histogram of Optical Flow (HOF), which was used by Laptev in [43] for human action recognition, and the Motion Boundary Histogram (MBH). The MBH descriptor was also proposed by Dalal et al. [14] for human detection, where the derivatives are computed separately for the horizontal and vertical components of the optical flow $I_\omega = (I_x, I_y)$. The spatial derivatives are computed for each component of the optical flow field I_x and I_y independently. After that, the orientation information is quantized into histogram, similarly to that for the HOG descriptor (8-bin histogram for each component). Finally, these two histograms are normalized separately with the L_2 norm and concatenated together to form the final representation. Since the MBH represents the gradient of the optical flow, constant motion information is suppressed and

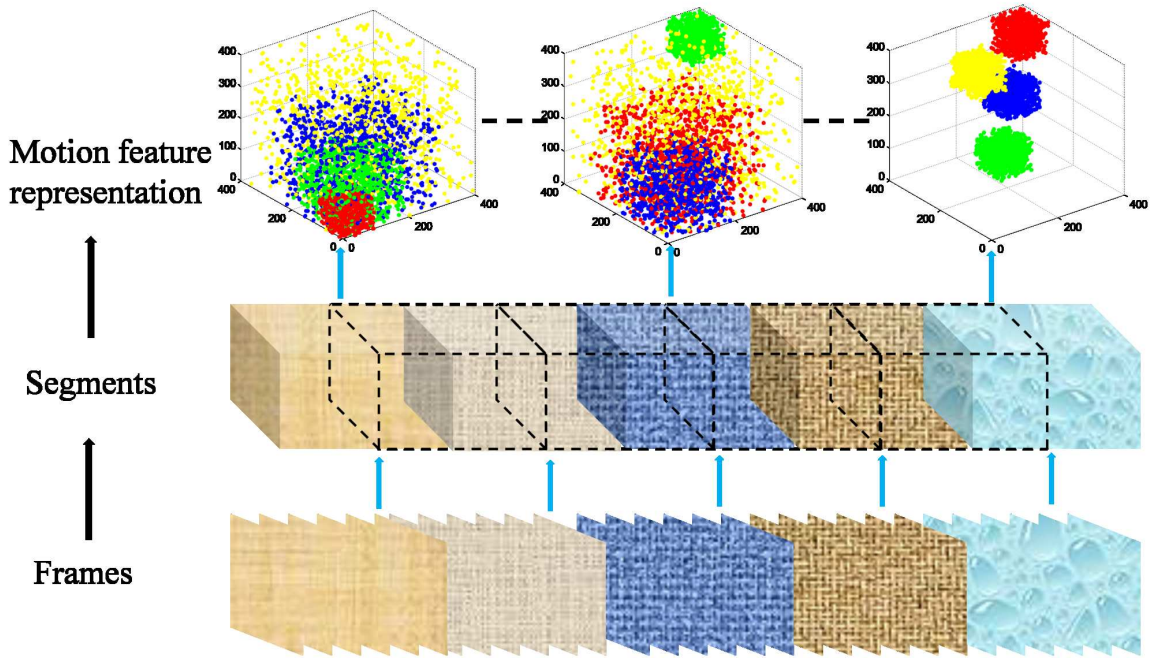


Fig. 3.1 Illustration of our segment-based approach. The original video is divided into segments by using non-overlapping and overlapping sampling (overlapped segment examples are drawn in dashes). After that, the feature representation is separately calculated for each segment. This figure is best viewed in color.

only the information concerning the changes in the flow field (i.e., motion boundaries) is kept.

According to the author [89], the MBH descriptor is the best feature descriptor for dense trajectories. One interesting property of the MBH is that it can cancel out camera motion. That is why it shows significant improvement on realistic action recognition dataset compared to other trajectory descriptors. We only use the MBH descriptor in this study to test the performance of our proposed segment-based method.

3.3.2 Segment-based Approach for Motion Feature

Our proposed segment-based approach is as follows. At first, the video is divided into fixed length segments. We choose different segment lengths to pick the optimal one. In particular, we choose segment lengths of 30, 60, 90, 120, 200 and 400 seconds. The lengths of 120 and 60 seconds are respectively close to the mean (115 s) and geometric mean (72 s) length of

the training dataset. The geometric mean value is also considered because it can eliminate the influence of outline cases, i.e., videos of exceptionally long durations. After that, the dense trajectory features are extracted from the entire segment. A "bag-of-words" approach is used to generate the final representation for each segment from the raw trajectory features (Fig. 3.1).

For the previous segment-based approach, a video is divided into continuous segments. This means information about the semantic boundary of a segment is not taken into account. However, this information is important because it keeps the semantic meaning of each segment. The simplest way to overcome this drawback is to use a denser sampling such as the overlapped segments. We use an overlapping strategy for the same segment length as in the non-overlapping experiments. In practice, we use uniform segment sampling with 50% of overlapping. This means the number of segments will be doubled for each overlapping experiment.

Another way to extract segments with boundary information is to employ a shot boundary detection technique. For a fast implementation, we use the algorithm proposed in [22]. This technique is also used in the Nikon 2010 MED system [54]. Basically, at first, this method constructs a space-time image from the input video. We can sample points or calculate the color histogram for each frame to construct the space-time image. This will reduce the 2D frame image to the space dimension of the space-time image. The time dimension is the number of frames of the video. The Canny edge detection algorithm is used to detect the vertical lines after attaining the space-time image. Each detected vertical line is considered as a scene cut. The method in [22] also proposed solutions for other kinds of scene transitions such as a fade or wide. However, from our previous study, this method showed poor results in these cases. Thus, we only adopted the scene cut detection algorithm. Each detected scene cut is considered a segment in our experiments.

Our proposed segment-based approach is compared with the video-based one. Actually, when the segment length is long enough, it becomes the entire video. In that case, we can consider the video-based approach a special type of segment-based approach.

3.4 Experimental Setup

3.4.1 Dataset

We tested our method on TRECVID MED 2010 and TRECVID MED 2011 datasets. An event kit is provided with the definitions and textual descriptions for all the events for each dataset. The first dataset contains 3,468 videos, including 1,744 videos for training and 1,724 video clips for testing, containing a total of more than 110 video hours. In TRECVID MED 2010, there are 3 events classes: assembling a shelter, batting in a run, and making a cake. The TRECVID MED 2011 dataset defined the 15 event classes listed in Table 3.1. The first five events (E001-E005) are used for training and validation and the last 10 events (E006-E015) are used for testing. It comprises of over 45,000 video clips for a total of 1,400 hours of video data. All the video clips are divided into three sets: event collection (2392 video clips), development collection (10198 video clips), and test collection (31,800 video clips). It is worth noting that these two datasets contain a major number of background video clips, i.e., video clips that do not belong to any event. The number of positive videos in the event collection is also listed in Table 3.1.

3.4.2 Evaluation Method

Figure 3.2 shows our evaluation framework for the motion features. We conducted experiments using the proposed segment-based approach and the video-based approach for comparison. We use the library published online by the author[4] to extract dense trajectory feature. The source code is customized for pipeline processing using only an MBH descriptor to save computing time but other parameters are set to default. Due to the large number of features produced when using the dense sampling strategy, we use the "bag-of-words" approach to generate the features for each segment. At first, we randomly select 1,000,000 dense trajectories for clustering to form a codebook of 4000 visual codewords. After that, the frequency histogram of the visual words is computed over the videos/segments to generate the

[4]http://lear.inrialpes.fr/people/wang/dense_trajectories

Table 3.1 List of events and its number of positive samples in event collection set of MED 2011 dataset.

Event ID	Event Name	#Pos videos
E001	Attempting a board trick	173
E002	Feeding an animal	168
E003	Landing a fish	152
E004	Wedding ceremony	163
E005	Working on a woodworking project	159
E006	Birthday party	221
E007	Changing a vehicle tire	119
E008	Flashmob gathering	191
E009	Getting a vehicle unstuck	151
E010	Grooming an animal	143
E011	Making a sandwich	186
E012	Parade	171
E013	Parkour	134
E014	Repairing an appliance	137
E015	Working on a sewing project	124

final feature vector. We also adopt the soft assignment weighting scheme, which was initially proposed by Jiang in [31], to improve the performance of the "bag-of-words" approach.

Once all the features are extracted, we use the popular Support Vector Machine (SVM) for the classification. In particular, we use the LibSVM library available online[5] and adopt the one-vs.-rest scheme for multi-class classification. We annotate the data in the following way to prepare it for the classifier. All the videos/segments from positive videos are considered positive samples, and the remaining videos/segments (in the development set) are chosen as

[5]http://www.csie.ntu.edu.tw/~cjlin/libsvm/

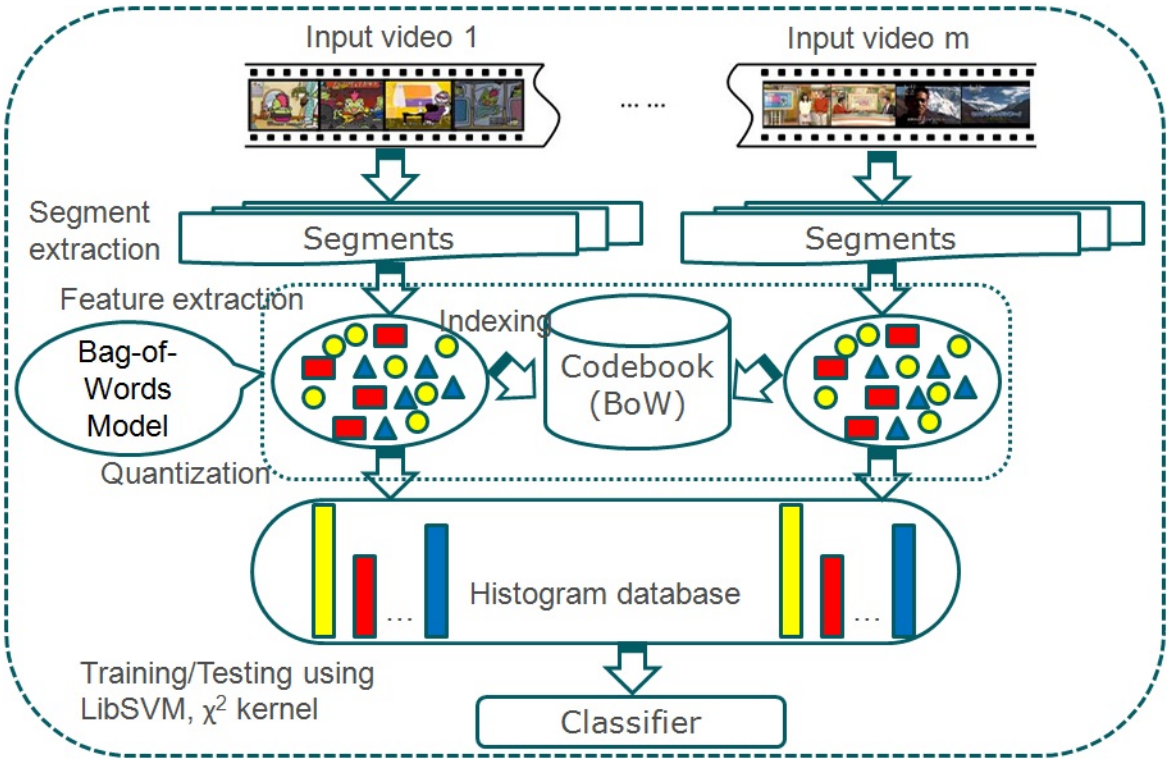


Fig. 3.2 Evaluation framework for our baseline MED system

the negative samples. For testing purposes, we also use the LibSVM to predict the scores of the videos/segments in each testing video. The score of a video is defined as the largest score among its videos/segments. This score indicates how likely a video belongs to an event class.

3.5 Experimental Result

This section presents the experimental results from using our proposed approach on the MED 2010 and MED 2011 dataset. We also present the results of combining various segment lengths using the late fusion technique. This is a simple fusion technique where the predicted score of each video is the average one of that video in all combined runs. We also report the performance of our baseline event detection system using the keyframe-based and video-based approach for comparison.

All the experiments were performed on our grid computers. We utilized up to 252 cores for the parallel processing using Matlab codes. All the results are reported in terms of the

Table 3.2 Results on the MED 2010 dataset using non-overlapping sampling.

Event/MAP	30 s	60 s	90 s	120 s	200 s	400 s	Late fusion
Assembling a shelter	0.4140	0.4511	0.4339	0.4457	0.4595	**0.4610**	0.4532
Batting in a run	0.7650	0.7852	0.7799	0.7553	0.7823	**0.7871**	0.7181
Making a cake	0.3596	0.3636	0.3433	0.3569	0.3058	0.3032	**0.3727**
All	0.5129	**0.5333**	0.5190	0.5193	0.5158	0.5171	0.5146

Table 3.3 Results on the MED 2010 dataset using overlapping sampling.

Event/MAP	30 s	60 s	90 s	120 s	200 s	400 s	Late fusion
Assembling a shelter	0.4177	**0.4781**	0.4617	0.4614	0.4601	0.4682	0.4486
Batting in a run	0.7727	0.7918	**0.7975**	0.7886	0.7893	0.7756	0.7691
Making a cake	0.4083	0.3819	0.3155	0.3415	0.3464	0.3239	**0.4232**
All	0.5329	**0.5506**	0.5249	0.5305	0.5319	0.5226	0.5470

Mean Average Precision (MAP). We calculate MAP using the TRECVID evaluation tool[6] from the final score of each video in the test set. The best performing feature is highlighted in bold for each event.

3.5.1 On TRECVID MED 2010

Non-overlapping and overlapping sampling

Table 3.2 lists the results from our segment-based approach when using a non-overlapping sampling strategy. These results show that the performance is rather sensitive to the segment length and it is also event-dependent. For example, the detection results of the first event, "assembling a shelter", are better when the segment length is increased. On the other hand, the

[6] http://www-nlpir.nist.gov/projects/trecvid/trecvid.tools/

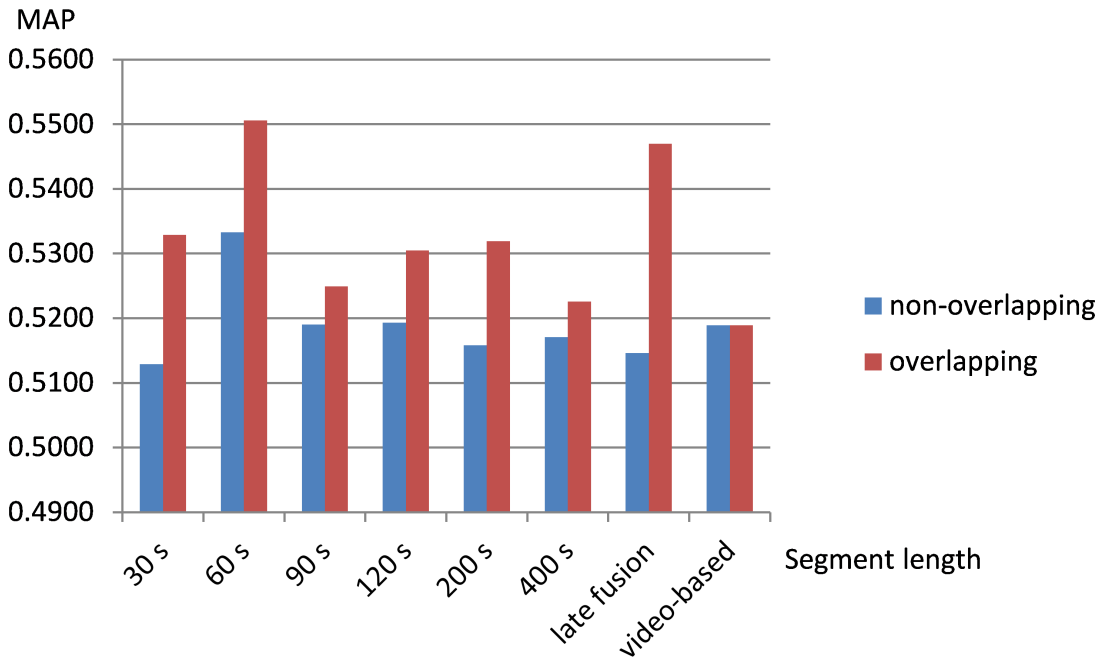


Fig. 3.3 Results from using segment-based approach with non-overlapping and overlapping sampling on MED 2010. In all cases, the overlapping sampling performs the best

"making a cake" event tends to be more localized, i.e. the shorter the segment, the better the performance. The performance of the "batting in a run" event is quite stable when segment length is longer than 60 s. However, it is decreased 2% at 30 s. This suggests that shorter lengths can harm the performance. In general, the performance of a 60-s segment is the best. This length is also around the geometric mean length of the training set. Thus, we got peak results for segment length around geometric mean point.

We further investigated the performance of a denser segment sampling, i.e., an overlapping sampling strategy. Interestingly, the MAP score in Table 3.3 is consistently increased for each event compared to the results without using overlapped segments. Figure 3.3 shows a detailed comparison between the two strategies in terms of the over-all performance. We again found that the performance with a segment length around the geometric mean length (60 s) was the best. We also combined the performances of all the segment lengths using late fusion and the results are listed in the last column of Tables 3.2 and 3.3. The late fusion strategy can benefit the "making a cake" event, but it decreased the performances of the remaining events. The overall performance is lower than the best one.

Table 3.4 Comparison of different segment-based approaches with the video-based approach on the MED 2010 dataset.

Event/MAP	Best non-overlapping	Best overlapping	SBD segments	Video-based
Assembling shelter	0.4511	0.4781	0.4284	**0.4911**
Batting in a run	0.7852	**0.7918**	0.7866	0.7902
Making a cake	0.3636	**0.3819**	0.1918	0.2755
All	0.5333	**0.5506**	0.4689	0.5189

Segment sampling based on shot boundary detection

The second column in Table 3.4 shows the performance when shot boundary detection is used to extract segments. Unexpectedly, the performance is quite low even when compared with the video-based approach (listed in the last column). There are two possible reasons for this low level of performance: (1) The shot boundary detection technique is inaccurate when used on uncontrolled capturing videos; (2) the shot units may not contain enough information to determine an event. The second reason suggests that combining multiple shots to form a segment may improve the performance. Thus, we have conducted a segment-based experiment based on this observation using segments extracted from multiple shots. However, we did not see any significant improvement. Thus, the first reason is why this experiment had poor result.

We also included the best results from the segment-based experiments using non-overlapping and overlapping sampling in Table 3.4 for comparison. In general, our segment-based approach outperforms the video-based approach by more than 3% in terms of MAP. We did not conduct a keyframe-based experiment because we learned that it is inefficient compared to the video-based approach.

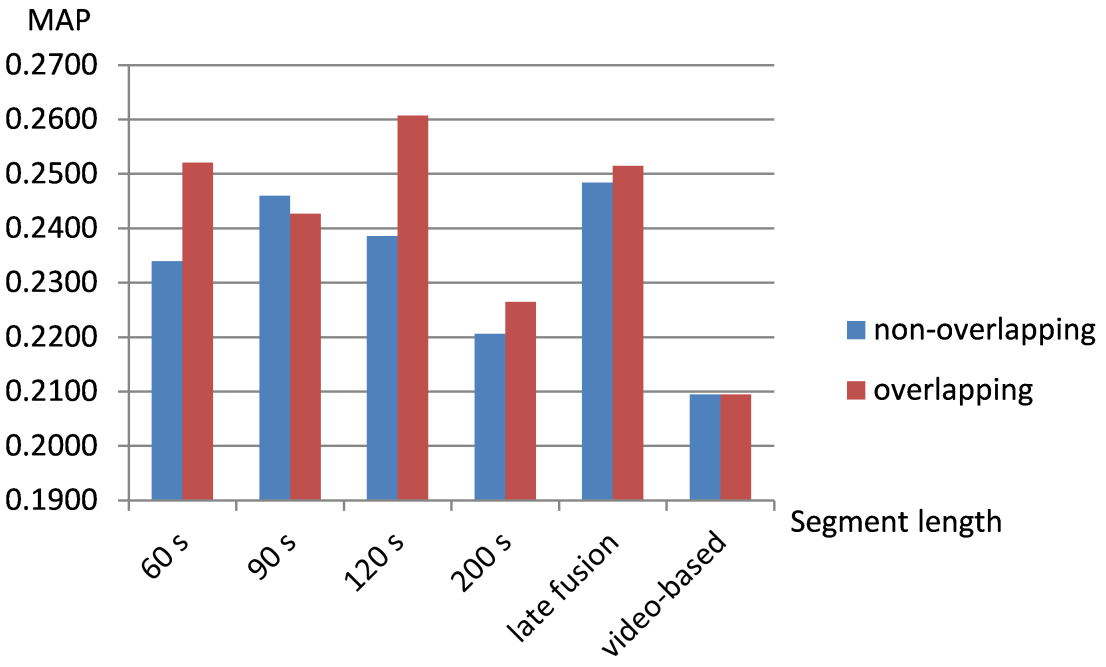


Fig. 3.4 Results from using segment-based approach with non-overlapping and overlapping sampling on MED 2011. In most cases, the overlapping sampling performs the best.

3.5.2 On TRECVID MED 2011

We conducted the same segment-based experiments on MED 2011. For both the non-overlapping and overlapping experiments, we chose segment lengths of 60, 90, 120, and 200 seconds and compare them with the video-based approach. A late fusion strategy is also used to combine the performances of different segment lengths. We did not conduct a shot boundary detection experiment because we showed that it is inefficient. Tables 3.5 and 3.6 list the performances of each event for non-overlapping and overlapping experiment, respectively. Figure 3.4 shows a better view for comparing the overall performance. The result from using video-based approach, which is 0.2095 MAP, is also included for comparison. In most cases, the overlapping sampling had better results than the non-overlapping sampling. In all cases, the segment-based approach also outperforms the video-based approach. The best improvement was about 5%, which was obtained at 120 s using an overlapping sampling. The late fusion run also confirms its effectiveness for some events, such as “Flash-mob gathering” and “Working on a sewing project”.

Table 3.5 Results on the MED 2011 dataset using non-overlapping sampling.

Event/ MAP	60 s	90 s	120 s	200 s	Late fusion
E006	0.1060	**0.1277**	0.1162	0.1005	0.1217
E007	0.1003	**0.1521**	0.1461	0.0539	0.1419
E008	0.4811	0.4923	0.4840	0.4508	**0.4975**
E009	0.2077	0.2072	0.1962	0.1860	**0.2145**
E010	0.0794	**0.0916**	0.0486	0.0854	0.0771
E011	**0.0943**	0.0698	0.0903	0.0703	0.0805
E012	0.3061	0.3560	0.3052	**0.3639**	0.3309
E013	0.5974	0.6030	0.5861	0.5941	**0.6033**
E014	0.2307	0.2008	**0.2772**	0.1723	0.2585
E015	0.1364	**0.1599**	0.1357	0.1284	0.1583
All	0.2340	0.2460	0.2386	0.2206	**0.2484**

The updated MED 2011 dataset has less number of training videos (See Table 2.3). We also verify the effectiveness of our approach on this dataset. We conduct experiments at different segment lengths including 8 s, 16 s, 32 s, 64 s, 128 s and 256 s. The overall performance is shown in the last group of Fig. 3.5.

3.6 Discussion

3.6.1 Optimal Segment Length

It is true that the lengths of the event segments are quite different, even for the same events. Therefore, the fixed length video segments are obviously not the optimal solution to describe the events. However, compared to the video-based approach, as shown in our experiments

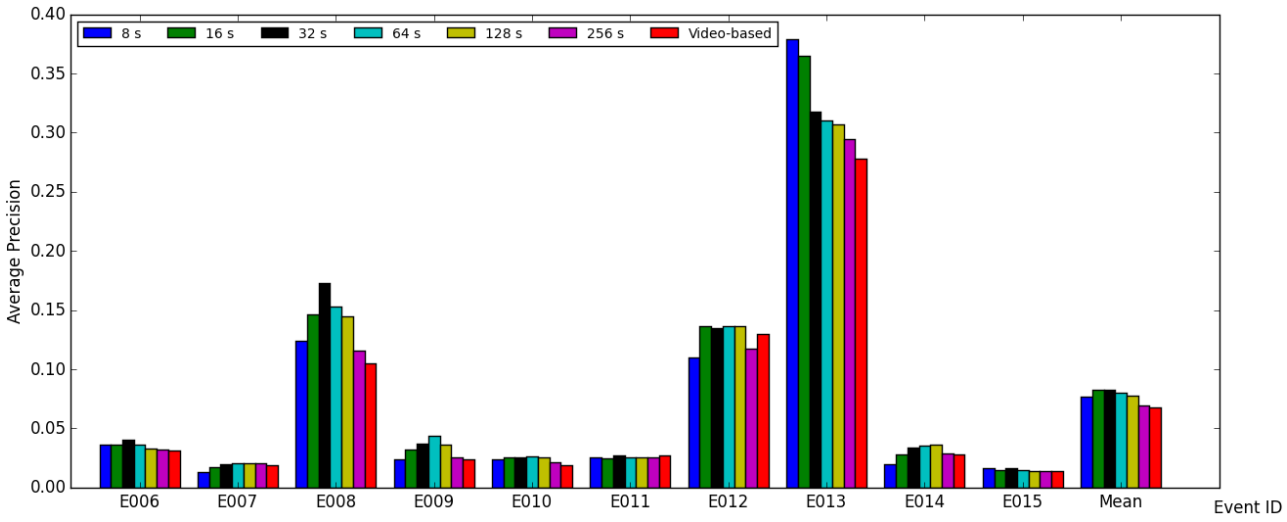


Fig. 3.5 Results from using segment-based approach with non-overlapping on the updated MED 2011 dataset.

Table 3.6 Results on the MED 2011 dataset using overlapping sampling.

Event/ MAP	60 s	90 s	120 s	200 s	Late fusion
E006	0.1074	0.1069	**0.1151**	0.1010	0.1086
E007	0.1570	**0.1733**	0.1552	0.1466	0.1610
E008	0.4788	0.4767	0.4969	0.4620	**0.4903**
E009	0.1830	0.1999	**0.2160**	0.1972	0.1954
E010	**0.1150**	0.0851	0.1008	0.0746	0.1108
E011	0.0602	0.0885	**0.1591**	0.0779	0.0819
E012	**0.3674**	0.3129	0.3150	0.3075	0.3293
E013	0.6025	0.5893	**0.6188**	0.5675	0.5872
E014	0.2718	0.2487	**0.2744**	0.2095	0.2706
E015	0.1777	0.1459	0.1562	0.1214	**0.1795**
All	0.2521	0.2427	**0.2607**	0.2265	0.2515

on the datasets of TRECVID MED 2010 and TRECVID MED 2011, the segment-based approach using overlapping strategy for extracting segments consistently outperforms.

It is ideal if the boundary of the event segment can be determined. However, this localization problem is difficult. The straightforward way to tackle this problem is extracting segments based on shot boundary information. This solution is reasonable because the event might be localized in certain shots. However, we obtained unexpected results due to the unreliability of shot boundary detection in uncontrolled video dataset and the event segment might span to several shots.

The method described in [30] suggests another approach to divide a video into segments. Instead of learning a randomized spatial partition for images, we can learn a randomized temporal partition for videos. However, this approach needs sufficient positive training samples while MED datasets have a small number of positive samples with large variation. On the other hand, it is also not scalable because learning and testing the best randomized pattern is time-consuming. Therefore, the fixed-length approach is quite simple but still effective.

Supposed the segment length is fixed, what is the optimal segment length for event detection? This is a difficult question and the answer depends on the dataset. The results of late fusion are quite close to the peak performance of each experiment. This suggests a methodical way to choose the optimal segment length, i.e., combining multiple lengths together (which is similar to [30]). However, to achieve the scalability, we should reduce the number of combined lengths as much as possible. From the experimental results on both the MED 2010 and MED 2011 dataset, we observed that with segment length from 60 s to 120 s, the performance is rather stable and close to the peak result. Interestingly, this range is approximate to the range from the geometric mean length to (arithmetic) mean length of the training sets. We also combined multiple segment lengths together using late fusion with equal weights for all segment lengths for comparison. There are two combined runs: one for segment lengths from 60 s to 120 s and the other is for all segment lengths. The result obtained when combining segment lengths from 60 s to 120 s is equivalent to the result obtained when combining all lengths, as shown in Table 3.8. Therefore, based on this

Table 3.7 Comparison of different segment-based approaches with the video-based approach on the MED 2011 dataset.

Event/MAP	Non-overlapping sampling			Video-based
	Best (at 90 s)	Late fusion (all lengths)	Late fusion (60, 90, 120 s)	
E006	**0.1277**	0.1217	0.1244	0.0959
E007	0.1521	0.1419	0.1369	0.1303
E008	0.4923	**0.4975**	0.4973	0.4766
E009	0.2072	0.2145	0.2064	0.0943
E010	0.0916	0.0771	0.0753	0.1020
E011	0.0698	0.0805	0.0813	0.0609
E012	**0.3560**	0.3309	0.3277	0.2858
E013	0.6030	0.6033	0.6096	0.5385
E014	0.2008	0.2585	0.2579	0.2138
E015	0.1599	0.1583	0.1622	0.0964
All	0.2460	0.2484	0.2479	0.2095

observation, we can choose the first combined run as an efficient way for solving the optimal segment length problem of the proposed segment-based approach on other datasets.

3.6.2 Scalability

For scalability, we discuss the storage and computation costs of our experiments. At first, our system does not consume a lot of disk storage because we only store the final representation of the videos or segments, not the raw features. We calculated the BoW features directly from the raw feature outputs using a pipeline reading technique. One drawback is that this technique requires a lot of memories. However, we handled this problem by encoding the raw features into smaller chunks and aggregating them to generate the final representation. By this way, we can manage the mount of memory usage.

In our framework, the most time-consuming steps are the feature extraction and representation (using the bag-of-words model). It is worth noting that the computation time for one video is independent of the segment length, which means our segment-based approach

Table 3.8 Comparison of different segment-based approaches with the video-based approach on the MED 2011 dataset.

Event/MAP	Overlapping sampling			Video-based
	Best (at 120 s)	Late fusion (all lengths)	Late fusion (60, 90, 120 s)	
E006	0.1151	0.1086	0.1083	0.0959
E007	0.1552	0.1610	**0.1616**	0.1303
E008	0.4969	0.4903	0.4871	0.4766
E009	**0.2160**	0.1954	0.1958	0.0943
E010	0.1008	0.1108	**0.1109**	0.1020
E011	**0.1591**	0.0819	0.0845	0.0609
E012	0.3150	0.3293	0.3341	0.2858
E013	**0.6188**	0.5872	0.5910	0.5385
E014	**0.2744**	0.2706	0.2694	0.2138
E015	0.1562	0.1795	**0.1795**	0.0964
All	**0.2607**	0.2515	0.2522	0.2095

has the same computational cost as the video-based approach. On the other hand, when we do experiments at the segment level, we will have more training and testing samples than that in the video-based approach. Thus, it will cost more in time to train and test using the segment-based approach. However, this cost is relatively small compared with the feature extraction and representation cost. For example, when using a grid computer with 252 cores, it took us about 10 hours to generate the feature representation for each segment-based experiment on MED 2010 dataset. In the mean time, we used one-core processor for the training and testing, but it only took about 4-8 hours for the training and 2-4 hours for the testing on each event. For the MED 2011 dataset, the computational cost was around 13 times bigger than the MED 2010 (linearly to the number of videos it contains).

3.7 Conclusion

We proposed using the segment-based approach for multimedia event detection in this work. We evaluated our approach by using the state-of-the-art dense trajectories motion feature on the TRECVID MED 2010 and TRECVID MED 2011 datasets. Our proposed segment-based approach outperforms the video-based approach in most cases when using a simple non-overlapping sampling strategy. More interestingly, the results are significantly improved when we using the segment-based approach with an overlapping sampling strategy. Therefore, the effectiveness of our methods on realistic datasets like MEDs is confirmed.

A segment-based approach with an overlapping sampling strategy shows promising results. This suggests the importance of segment localization on the MED performance. Suppose the segment length is fixed, we are interested in determining which segment is the best representative for an event. In this study, we also observed that the detection performance is quite sensitive to the segment-length and it depends on the dataset. The results obtained from the late fusion strategy is quite stable and close the peak performance. This suggests a methodical way to generalize the segment-based approach to other datasets. However, this method is not scalable because it requires a lot of computation costs. Therefore, learning an

optimal segment length for each event can be beneficial for an event detection system. This is also an interesting direction for our future study.

Chapter 4

Event Detection Using Sum-max Feature Aggregation

A clay pot sitting in the sun will always be a clay pot. It has to go through the white heat of the furnace to become porcelain.

– Mildred W. Struven

4.1 Introduction

The problem of aggregating low level representation into a higher level one has been well studied for image representation. Basically there are two main strategies to aggregate local image descriptors: sum pooling [36] and max pooling [79]. To understand about these pooling strategies, it is better to mention them in the context of bag-of-word model [12]. In this model, at first a dictionary or codebook with around thousands of codewords is trained using an unsupervised method such as K-means or Approximate K-means. After that, local features, which are often extracted using a standard SIFT [50] feature, are quantized into the codebook based on their distances to the nearest codewords. Finally, features that are assigned to a codeword are pooled to get a representative value for that codeword. The sum

Fig. 4.1 Example video for "assembling a shelter" event in the TRECVID MED 2010 dataset. The top row shows the relevant frames while the bottom row shows the noisy frames.

pooling technique simply takes a sum over responses to a visual word. This technique is useful when most of the features are relevant. On the other hand, the max pooling technique only select the largest value between features responding to a visual word. This technique only useful when at least one local feature is sufficiently discriminative. In this case, most of the remaining features can be irrelevant.

Sum pooling and max pooling techniques can be easily adopted for video representation. In this case, we can treat spatial-temporal local features in video as local features in image and apply the same framework. State of art performance can be obtained using bag-of-words model with the sum pooling technique in simple video classification/recognition tasks such as sports action videos [76] or studio setting movies [51]. This is due to the fact that discriminative features exist in the entire video in these datasets. However, this observation is not true on complex video datasets where the discriminative features may exist within a small part of the video. One example of these datasets is the TRECVID Multimedia Event Detection (MED) dataset[1], where most videos are captured by internet users and it tends to be noisy. Example of such noisy video is shown in Fig 4.1. In this case, video pooling for event recognition is much more challenging.

We are interested in the problem of video pooling for a more robust video representation. We consider a video as a layered structure where the lowest layer are frames, the top layer is the entire video, and the middle layers are the sequences of consecutive frames or the concatenation of lower layers. Based on this layered structure of video, we propose to use the sum-max video pooling to deal with noisy information in complex videos. Basically,

[1] http://www.nist.gov/itl/iad/mig/med10.cfm

we apply sum pooling at the low layer representation while using max pooling at the high layer representation. Sum pooling is used to keep sufficient relevant features at the low layer, while max pooling is used to retrieve the most relevant features at the high layer, therefore it can discard irrelevant features in the final video representation.

Our work is most related to [74], in which they proposed a segment-based approach to generate segment level representation using the sum pooling technique. Here we focus on different pooling techniques to generate the video representation. Experimental results on the TRECVID Multimedia Event Detection 2010 dataset shows the effectiveness of our method.

The rest of this chapter is organized as follows. Section 2 introduces the layered structure of video. Section 3 presents our sum-max pooling technique based on this layered structure. The experimental setup and experimental results are described in Section 4. Finally, Section 5 concludes this chapter with discussions on our future work.

4.2 Layered Structure of Video

As mentioned in the previous section, pooling over the whole video is not effective for complex video representation because these videos can contain irrelevant information. The direct solution to remove these irrelevant information from the final video representation is to pool over the relevant parts only. However, it is also non-trivial to determine which parts of the video are relevant or not.

The layered structure of video is a simply way to lessen the impact of irrelevant information. We define this layered structure as follows. The lowest layer are the frames of that video. The top layer is the entire video. The middle layers are the sequences of consecutive frames or the concatenation of lower layers. Figure 4.2 illustrates the layered structure in videos.

For the sake of simplicity, we only use one middle layer and the frame sequences in the middle layer are referred as segments in the rest of this chapter. In implementation, we choose the length of the segments varies in the following range: 15, 30, 45, 60, 75, 90, 105,

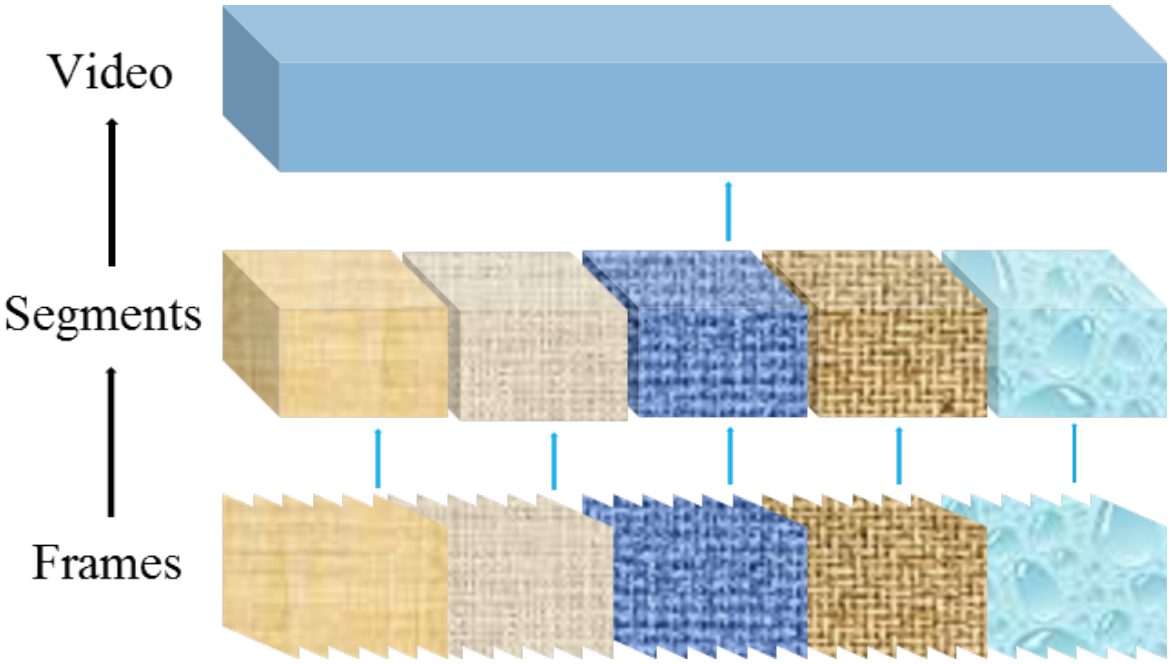


Fig. 4.2 Illustration of layered structure of video.

120, 135, 150, 165, 180, 195 and 210 seconds. We report the best segment length in Section 4.4.

4.3 Sum-max Video Pooling

Our sum-max video pooling method is proposed based on the layered structure of video and consists of two steps: (1) Applying sum pooling to aggregate features from all frames of each segment to generate the feature representation of that segment; (2) Applying max pooling to aggregate the segment-level features to form the video representation. The max-sum video pooling can be obtained in the same way but different in that max pooling is applied first, then the sum pooling. It is worth noted that, sum video pooling and max video pooling are two special cases when we applying sum-max video pooling and max-sum video pooling for the whole video respectively. Examples of sum-max and max-sum video pooling are shown in Fig 4.3.

In the context of bag-of-words model, suppose that there are N local descriptors in the video, each descriptor is denoted at $x_n \in R^D$, where n = 1,...,N and D is the feature

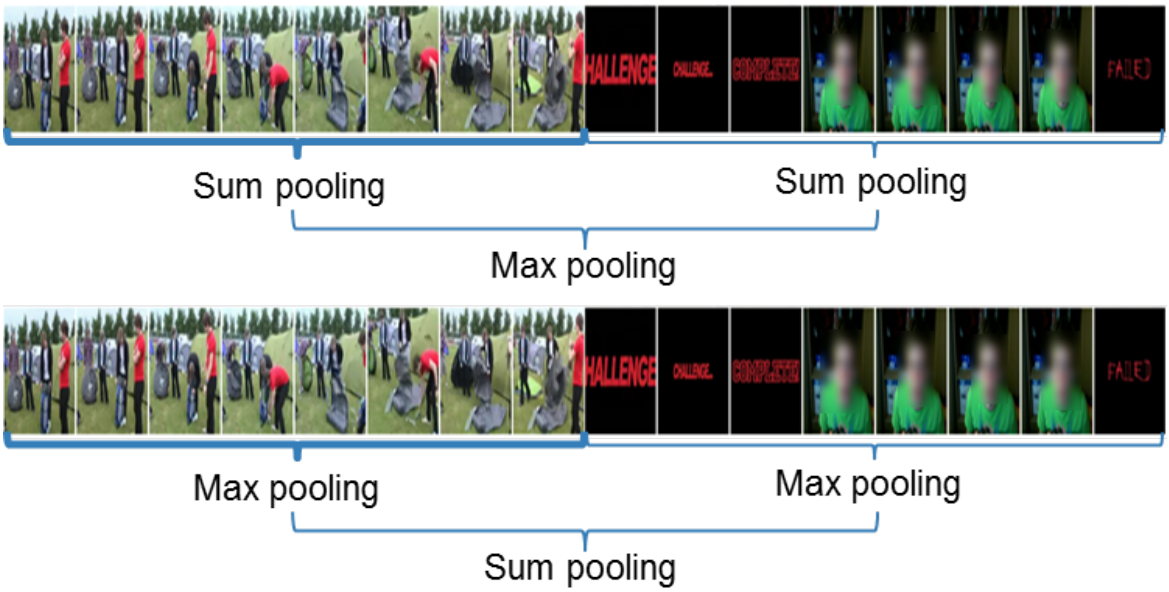


Fig. 4.3 Example of applying sum-max video pooling (top) and max-sum video pooling (bottom) methods on an "assembling a shelter" event video. It can be seen from the top image that after applying max pooling at the segment level, only relevant frames are encoded in the final representation.

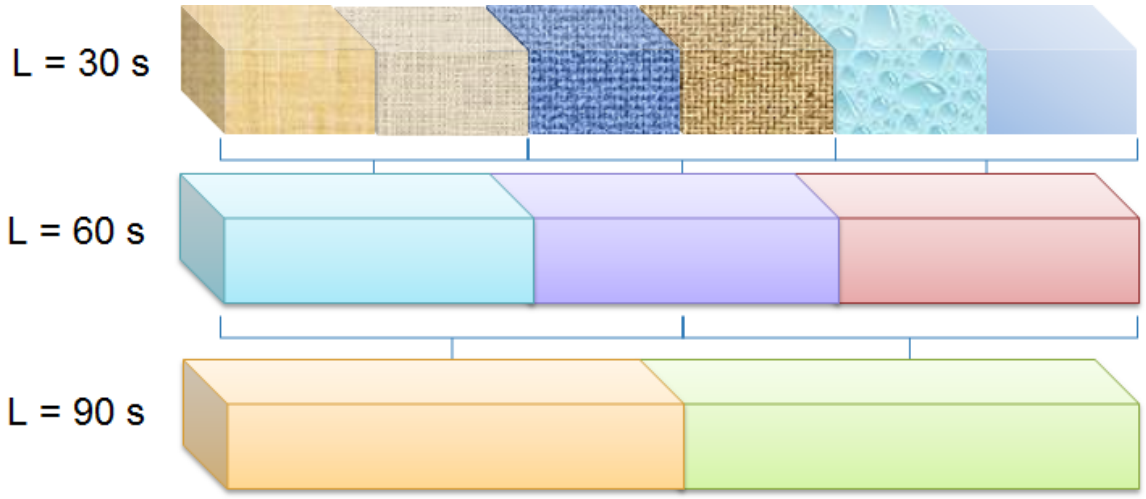


Fig. 4.4 Features from higher layers can be obtained from lower layers efficiently.

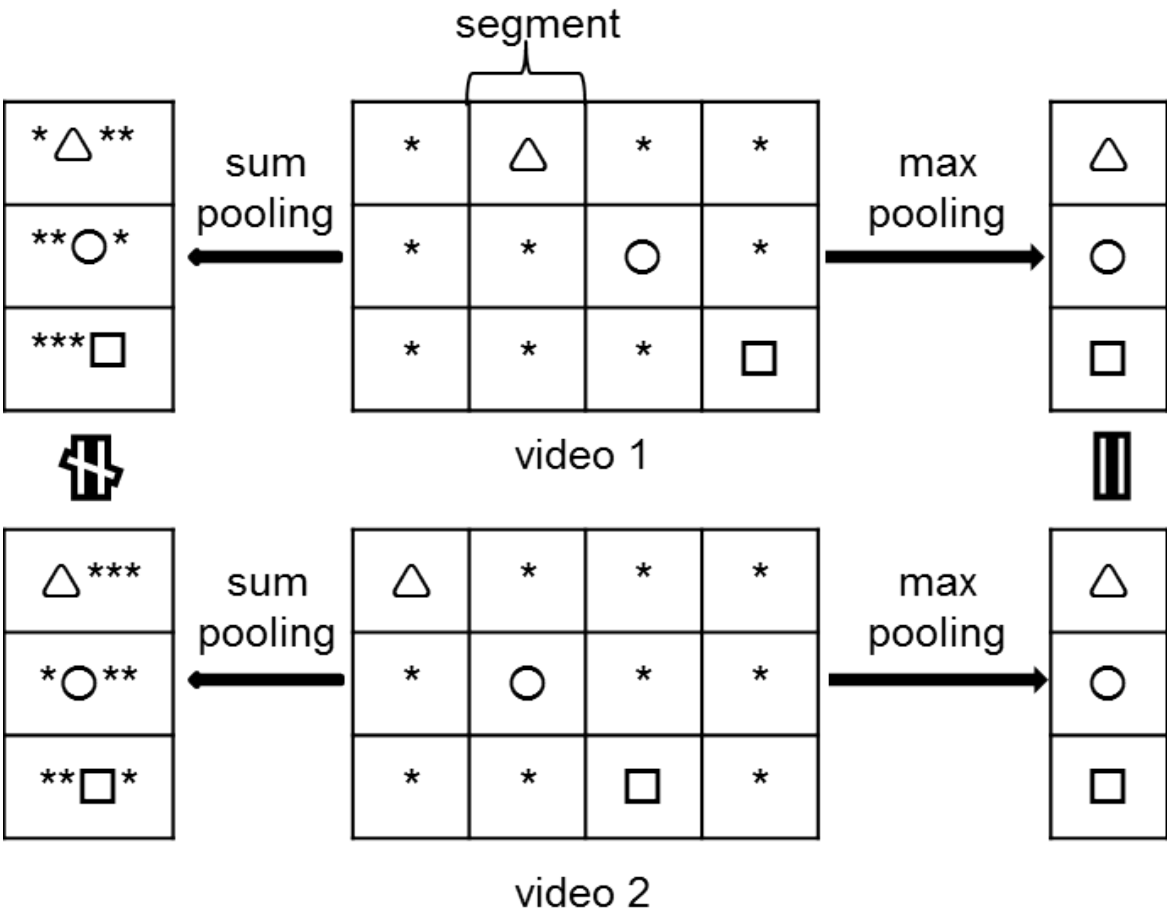


Fig. 4.5 Illustration of sum-max video pooling. △, O, □ represent relevant information; * represents different kinds of irrelevant information, which is popular in complex event data. Due to the native of the data, relevant information can appear in any part of the video, and can follow some temporal order.

dimension. Denote each visual word $m_k \in R^D$, where k = 1,...,K with K is number of visual words. $M = \{m_k\}$ is the set of visual words. The mid level coding of each descriptor can be expressed as $\phi_n = [\Phi_{1n}, ..., \Phi_{Kn}]$. Further suppose that the video contains S segments. Denote N_s is the number of local descriptors in segment s. The sum-max and max-sum video pooling at each visual word can be defined as follows:

$$\psi_k = Max_{s \in S}(\sum_{n \in N_s} \Phi_{kn}) \tag{4.1}$$

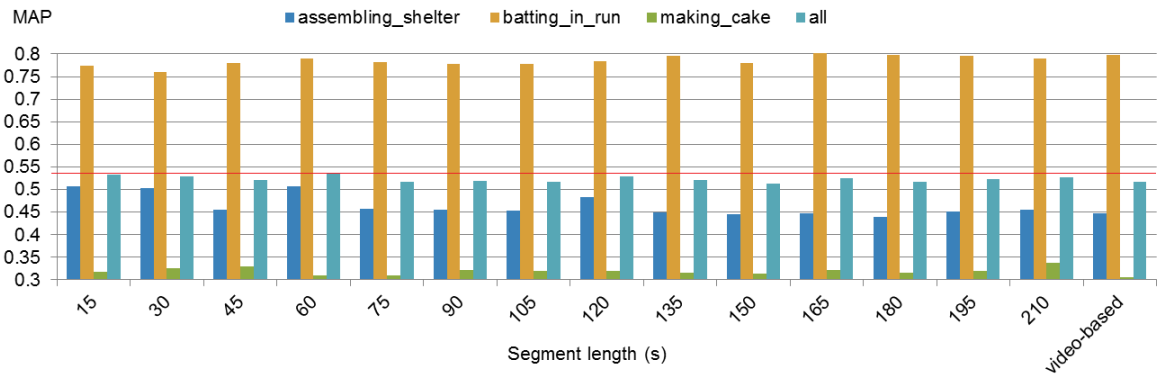


Fig. 4.6 Results on the MED 2010 dataset using the sum-max pooling technique at different segment lengths.

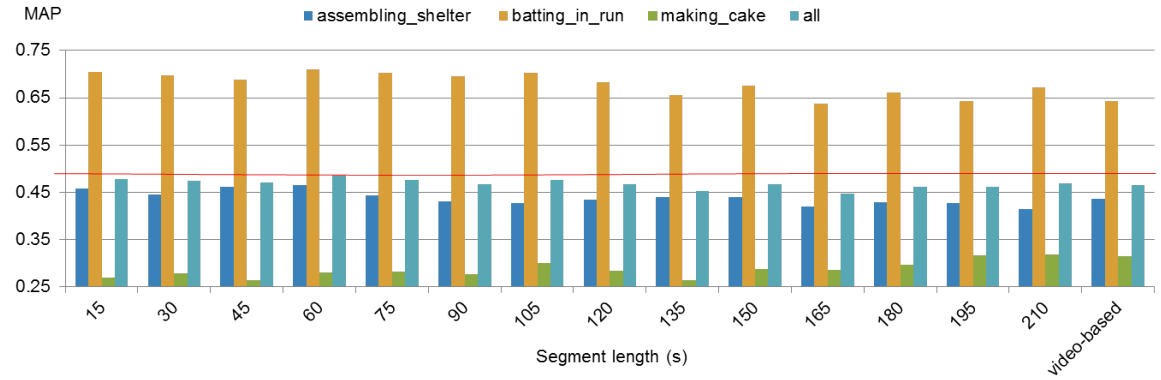


Fig. 4.7 Results on the MED 2010 dataset using the max-sum pooling technique at different segment lengths.

$$\psi_k = \sum_{s \in S} (Max_{n \in N_s} \Phi_{kn}) \tag{4.2}$$

An intuitive example of sum-max pooling is shown in Fig 4.5. As we can see, max pooling reserves the relevant information because noisy data tend to be varied, and none of any kind of them is dominant. In the contrast, sum pooling incorporates both relevant and irrelevant ones. Therefore, it is less representative than max pooling.

It is also worth noted that features from higher layers can be obtained from lower layers efficiently. In fact, we only need to extract feature one time. An illustration of features calculated from different segment lengths can be seen on Fig. 4.4.

4.4 Experiment

4.4.1 Experimental Setup

We tested our method on TRECVID MED 2010 dataset. An event kit is provided with the definitions and textual descriptions for all the events for each dataset. The first dataset contains 3,468 videos, including 1,744 videos for training and 1,724 video clips for testing, containing a total of more than 110 video hours. In TRECVID MED 2010, there are 3 event classes: *assembling a shelter* (E001), *batting in a run* (E002), and *making a cake* (E003).

We adopt the popular bag-of-words model to build our event recognition framework. At first, we use dense trajectory motion feature published by Wang [89] to calculate raw motion features as local trajectory descriptors. The library to extract these features is published online by the author[2]. The source code is customized for pipeline processing using only Motion Boundary Histogram (MBH) descriptor to save computing time but other parameters are set to default.

In the coding step, we randomly select 1,000,000 dense trajectories for clustering to form a codebook of 4000 visual codewords. After that, the frequency histogram of the visual words is computed over each segment to generate the feature vector for that segment. Finally, we apply the sum-max pooling technique as described in Section 4.3 to obtain the final video representation. We also adopt the soft assignment weighting scheme [31] with 5 nearest neighbors to improve the performance of the "bag-of-words" approach.

In the learning and testing step, we use the popular Support Vector Machine (SVM) for event classification. In particular, we use the LibSVM library available online[3] and adopt the one-vs.-rest scheme for multi-class classification.

[2]http://lear.inrialpes.fr/people/wang/dense_trajectories
[3]http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Table 4.1 Performance comparison of different video pooling strategies on the MED 2010 dataset.

Event/MAP	Max pooling	Sum pooling	Max-sum pooling (at 60 s)	Sum-max pooling (at 60 s)
E001	0.4365	0.4468	0.4646	**0.5072**
E002	0.6434	**0.7988**	0.7103	0.7900
E003	**0.3144**	0.3053	0.2806	0.3100
All	0.4648	0.5170	0.4852	**0.5357**

4.4.2 Experimental Result and Analysis

On the MED 2010 dataset

We report the results in terms of the Mean Average Precision (MAP). Results of sum-max video pooling and max-sum video pooling are showed in Fig 4.6 and Fig 4.7 respectively. Sum-max pooling improves the overall performance, especially for "assembling a shelter" event. The best performance is obtained at the segment length of 60 s (same as observed in [74]). Max-sum video pooling did not achieve good results compared to sum-max video pooling. The reason for the low performance of max-sum pooling can be due to the lost of relevant information when max-pooling is applied first.

We also observed that the performance largely depends on the segment length and the event itself. For example, we can get better performance with short segment lengths for the event "assembling a shelter", while the event "making a cake" tends to have better performance with longer segments.

We summarize our experimental results in Table 4.1. The best performing feature is highlighted in bold for each event. In general, pooling over segments is more effective, i.e, sum-max pooling outperforms sum pooling and max-sum pooling outperforms max pooling. In the best case, sum-max video pooling outperforms the traditional sum pooling up to 2% in terms of MAP.

On the MED 2011 dataset

In this experiment, we calculate the sum-max video pooling at different segment lengths including 8 s, 16 s, 32 s, 64 s, 128 s and 256 s. Results of our proposed methods on MED 2011 dataset are shown on Fig. 4.8 and Fig. 4.9 respectively. Our best results are obtained at 8 s when using χ^2 SVM and 32 s when using linear SVM.

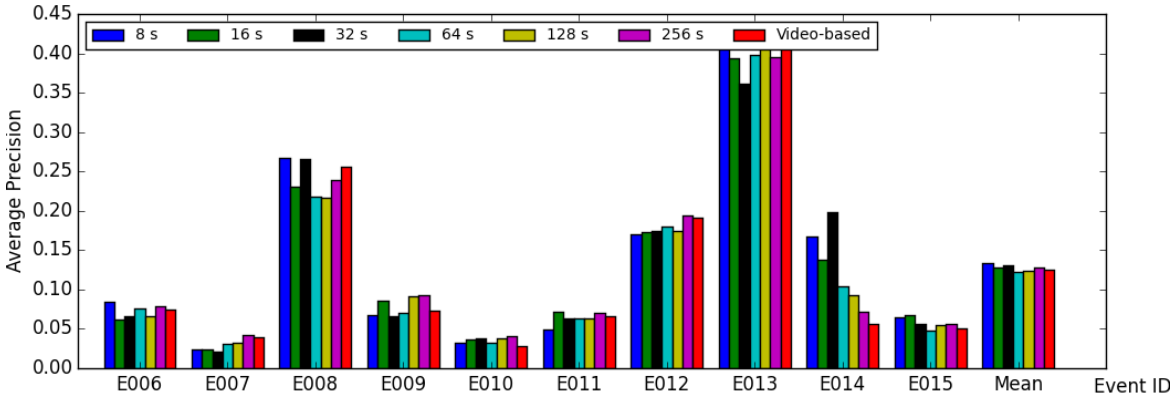


Fig. 4.8 Results on the MED 2011 dataset using the sum-max pooling technique at different segment lengths (χ^2 SVM).

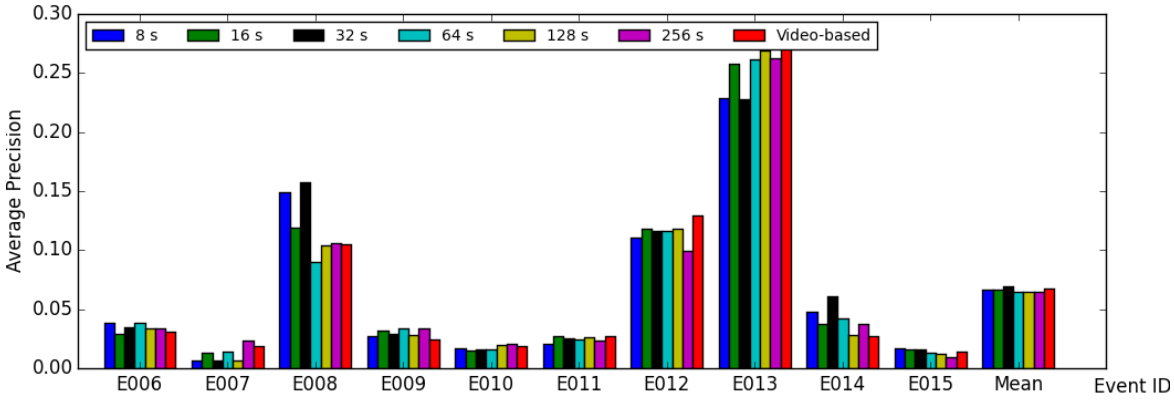


Fig. 4.9 Results on the MED 2011 dataset using the sum-max pooling technique at different segment lengths (linear SVM).

4.5 Conclusion

We proposed to use a sum-max video pooling technique to combine both sum pooling and max pooling into a holistic video representation. This pooling technique is based on the

layered structure of video. Preliminary results showed that this is an promising direction for video representation.

One limitation of the current approach is that the performance depends on the segment length. Therefore, we suggest to investigate a better approach to utilize the layered structure of video for video representation.

For video representation, temporal information is also very important. However, it is difficult to encode temporal information because video lengths are very varied. Therefore, exploring temporal pooling for video representation is also a good research direction.

Chapter 5

Event Detection Using Event-Driven Multiple Instance Learning

You never change things by fighting the existing reality. To change something, build a new model that makes the existing model obsolete.

– Buckminster Fuller

5.1 Introduction

The problem of recognizing complex event in videos has become a popular research topic due to the explosive growth of video data. A complex event can involve several actions or activities and happens in some particular settings. Therefore, recognizing complex event is more challenging than single action recognition. However, most complex detection systems are still based on the techniques that was developed for action recognition [69, 90]. These methods basically extract and aggregate local feature descriptors from the whole video to create a unique video representation. This strategy might be not effective for complex event detection because it treats different parts of the video equally. Therefore, it neutralizes the important local information of an event.

Fig. 5.1 Event "Grooming an animal" in the TRECVID MED 2012 dataset. The event kit includes example videos and an event description which provides valuable cues to detect that event.

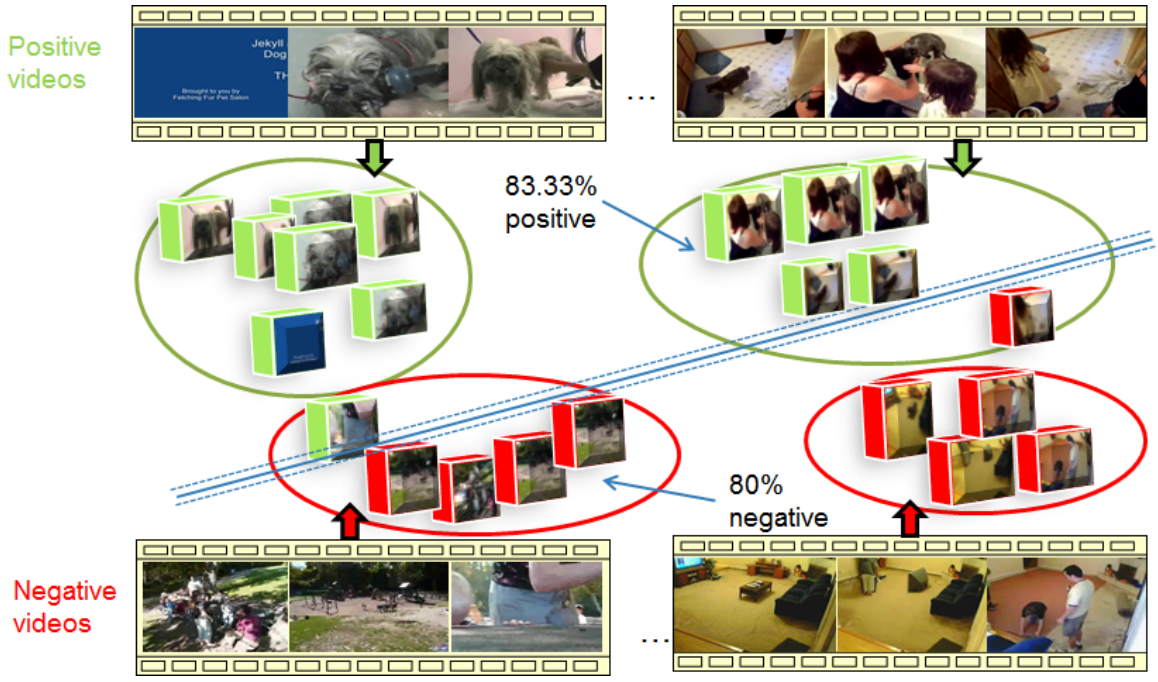


Fig. 5.2 Illustration of pSVM [38] method. Different from both miSVM and MISVM solutions, pSVM allows some positive instance in the negative bags, which is suitable for real videos.

In practice, human can recognize a complex event by spotting several evidences in video [5]. This paper also demonstrated that better performance can be obtained by leveraging positive and negative visual cues selected by humans. Therefore, it is important to automatically detect key evidences for event detection. Several researchers have been working on this direction. Tang *et al.* [84] split the video into segments and models key segments and its duration as latent variables. Vahdat *et al.* [86] focus on intra-class variation by localizing only the most salient evidence using latent SVM. Lai *et al.* [39] detect salient instances in video based on a variant multiple instance learning, which was proposed by Yu *et al.* [95]. In another work [38], they represent static and dynamic instances as sparse features and adopt a learning-to-rank strategy to detect key evidence. In general, these approaches are based on the assumption that segment annotation can be obtained from its video label. However, this is a weak assumption because the importance of each segment is not taken into account.

On the other hand, the importance of a segment to an event can be obtained by matching its concept-based representation against the evidential description of that event. Some works

have been using the event description for zero-shot event detection such as in [10, 92]. To the best of our knowledge, no work has taken into account this information for detecting key evidence in videos. However, the evidential description of an event provides valuable information to detect that event. Example of an event description (excerpted) is shown in Fig. 5.1.

Motivated by this observation, we propose a new method, Event-driven Multiple Instance Learning (EDMIL), to learn key evidences for complex event detection. We treat each segment as an instance and model it in a multiple instance learning framework [2], where each video is a "bag". The instance-event similarity is quantized into different levels of relatedness. Intuitively, the most (ir)relevant instances should have higher (dis)similarities. Therefore, we propose to learn the instance labels by jointly optimize the instance classifier and its related level. We evaluate our proposed method on the large scale TRECVID MED 2012 dataset. Comparing to other instance-based learning methods such as [2, 39], our method achieves a superior performance.

The remaining of this chapter is organized as follows. In the next section, we present the method to calculate the instance-event similarity. Our proposed solution is introduced in Section 5.3. The experiments and results are shown in Section 5.4. Finally, Section 5.5 concludes this work.

5.2 Instance-Event Similarity

In order to calculate the similarity between an instance and an event, we adopt a concept expansion strategy as in [10]. Our method is similar in spirit, however, we apply at instance level which is more accurate. The outline of our method is illustrated in Fig. 5.3 and it consists of four steps.

Step 1: Concept detection. We use the concept collection that proposed in [99] to cover a wide range of concept that can appear in realistic videos. This collection contains C = 1183 categories including 205 scene categories from the Places Database[99] and 978 object categories from the ImageNet 2012[16]. The concept detection part is done by using the

provided pre-trained model[1]. To detect concept for the whole segment, we detect concept at sample frames and make the average aggregation.

Step 2: Event representation. We use standard natural language processing techniques to create the text-based event representation. At first, the event description is pre-processed by removing stop words and lemmatizing. It is then converted into a bag-of-words representation, where the dictionary is obtained from the English Wikipedia corpus. Tf-idf weighting scheme is also employed to put a higher weight on frequent as well as rare words.

Step 3: Concept-event similarity. To resolve the mismatch between words in the concept collection and event description, we adopt the concept expansion strategy [10]. For each concept category, we add the 10 most similar concepts obtained from word2vec[58] model[2] to expand this category. It is then represented by a bag-of-words vector with tf-idf weights. Based on this representation, we can calculate the cosine similarity s_c^e between each concept category and the event description. Table 5.1 shows top five most relevant concepts for some events on the MED 2012 dataset.

Step 4: Instance-event similarity. Having obtained the concept score x_c at each segment and the concept-event similarity as in Step 1 and Step 3, the instance-event similarity is calculated using the cosine similarity:

$$S_i^e = \frac{\sum_{c=1}^{C} s_c^e x_c}{\sqrt{\sum_{c=1}^{C} (s_c^e)^2}\sqrt{\sum_{c=1}^{C} (x_c)^2}}, \tag{5.1}$$

5.3 Event-Driven Multiple Instance Learning

5.3.1 Problem Formalization

Suppose we have V training videos, and I_v instances in video v. We can calculate the similarity S_{iv}^e between an instance iv to a particular event e using Eq. (5.1). Suppose there

[1] http://places.csail.mit.edu
[2] https://code.google.com/p/word2vec

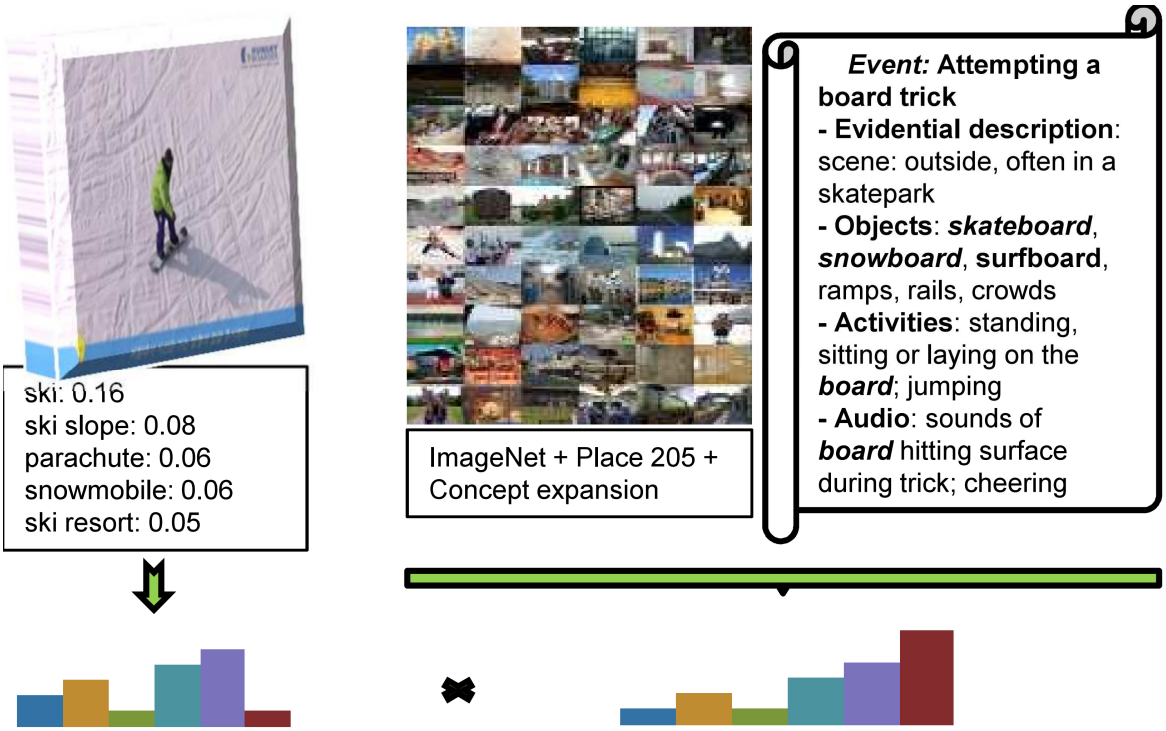


Fig. 5.3 Outline of our method to calculate the instance-event similarity. Note that the concept expansion technique can bridge concept "ski" in the instance segment to the evidential description.

Table 5.1 Top five concepts discovered by our system for 25 events in the MED 2012 dataset.

Event ID	**Top five importance concepts discovered by our system**
E001	Ski, slide rule, ski resort, ski mask, ice skating rink
E002	Meat loaf, white shark, food court, pop bottle, cleaver
E003	Anemone fish, pole, raft, sturgeon, boat deck
E004	Groom, bridegroom, banquet hall, gown, altar
E005	Jigsaw puzzle, bamboo forest, carpenter's kit, thatch, wooden spoon
E006	Table lamp, lampshade, torch, candle, custard apple
E007	Recreational vehicle, car wheel, amphibian, scooter, sports car
E008	Monitor, chime, bell, whistle, ballroom
E009	Recreational vehicle, amphibian, tank, car wheel, motor scooter
E010	Nail, bathtub, shower, fur coat, washbashin
E011	Pizza, bagel, meat loaf, cheeseburger, vegetable garden
E012	Recreational vehicle, amphibian, tank, sports car, freight car
E013	Playground, volleyball, picnic area, sports car, table lamp
E014	Toaster, dish washer, washing machine, refrigerator, space heater
E015	Sewing machine, dragonfly, syringe, clothing store, construction site
E016	Digital watch, classroom, CD player, crossword, stopwatch
E017	Tray, game room, cassette player, CD player, waiting room
E018	Backpack, walking stick, pop bottle, sleeping bag, plastic bag
E019	Tile roof, mortar, nail, jigsaw puzzle, drumstick
E020	Ballpoint, pencil box, rubber eraser, quill pen, pencil sharpener
E021	Tricycle, mountain bike, scooter, bicycle-built-for-two, unicycle
E022	Toaster, refrigerator, dish washer, washing machine, space heater
E023	Schipperke, otter hound, bluestick, collie, Tibetan terrier
E024	Forest path, cellular telephone, phone booth, platform, dial phone
E025	Boxing ring, fairway, hand-held computer, bell cote, chime

is R level of relatedness from an instance to an event. We define two predict functions for positive and negative instances at level r as follows.

$$P_{pos}(S^e_{iv}, r) = \begin{cases} 1, & \text{if } Rank(S^e_{iv}) \leq r \\ -1, & \text{otherwise} \end{cases}, \text{ and} \tag{5.2}$$

$$P_{neg}(S^e_{iv}, r) = \begin{cases} -1, & \text{if } Rank(S^e_{iv}) \leq r \\ 1, & \text{otherwise} \end{cases}, \tag{5.3}$$

where $Rank(.)$ is the function to quantize a similarity into a related level. Note that smaller value of r results a higher confidence in the predict functions. We now learn the parameters of the instance classifier jointly with the related level r by optimizing the following objective function:

$$\begin{aligned} \min_{\mathbf{w},b,y,r} \frac{1}{2}\|\mathbf{w}\|^2 + C_f \sum_{v=1}^{V}\sum_{i=1}^{I_v} L_f\left(y_{iv}, \mathbf{w}^T\mathbf{x}_{iv} + b\right) \\ + C_p \sum_{v=1}^{V}\sum_{i=1}^{I_v} L_p\left(y_{iv}, P(S^e_{iv}, r)\right). \end{aligned} \tag{5.4}$$

C_f and C_p are cost parameters to control the influence of each loss function. Note that in the special case where $C_p = 0$, the above formulation becomes a classic large-margin problem. $L_f(.)$ and $L_p(.)$ are two loss functions that will be jointly minimized. The first loss function minimizes the loss due to the classification mismatch based on the instance feature. The second one minimizes the loss due to the prediction obtained from the prior knowledge. Intuitively, when the related level r increases, the first loss will also tend to increase while the second loss will become smaller, and vice versa. $L_f(.)$ and $L_p(.)$ can be any loss function. Throughout this work, we use the standard hinge-loss function for $L_f(.)$: $L_f\left(y_{iv}, \mathbf{w}^T\mathbf{x}_{iv} + b\right) = \max(0, 1 - y_{iv}(\mathbf{w}^T\mathbf{x}_{iv} + b))$, and the $L_p(.)$ function is defined so that it will penalize more on the high confident predictions:

$$L_p\left(y_{iv}, P(S^e_{iv}, r)\right) = \begin{cases} S^e_{iv}, & \text{if } P(S^e_{iv}, r) \neq y_{iv} \\ 0, & \text{otherwise} \end{cases}.$$

5.3.2 Optimization Procedure

The optimization problem in Eq. (5.4) is a mixed-integer program which is not convex. In order to solve this problem, we apply the alternating optimization strategy to search for a suboptimal solution:

1. Fix instance labels y_{iv} and solve for $\mathbf{w}$ and b. By fixing y_{iv}, the optimization problem becomes a classic SVM:

   $$\min_{\mathbf{w},b} \tfrac{1}{2}\|\mathbf{w}\|^2 + C_f \sum_{v=1}^{V}\sum_{i=1}^{I_v} L_f\left(y_{iv}, \mathbf{w}^T\mathbf{x}_{iv} + b\right).$$

   Thus it can be solved using a regular SVM solver.

2. Fix $\mathbf{w}$ and b, solve for r and update y_{iv}. The problem now becomes:

   $$\min_{y,r} C_f \sum_{v=1}^{V}\sum_{i=1}^{I_v} L_f\left(y_{iv}, \mathbf{w}^T\mathbf{x}_{iv} + b\right) + C_p \sum_{v=1}^{V}\sum_{i=1}^{I_v} L_p\left(y_{iv}, P(S^e_{iv}, r)\right).$$

   We propose a greedy strategy to solve for this problem. At first, we iterate through all level of relatedness to search for the optimal r by finding the minimum total loss when updating y_{iv} using Eq. (5.2, 5.3). Because the most positive and negative instances will be selected first, there will be a higher possibility to correct mismatched labels that were learned in the previous step. Lastly we update instance labels using Eq. (5.2, 5.3) with the optimal r.

Because this is not a convex optimization problem, the initialized values of y_{iv} should be carefully selected. To this end, we use the same initilization method as in [2, 39], where instance labels are same with its "bag" (video) label.

It is also worth noted that the optimization framework only keeps updating the instance labels while the instance features are unchanged. Thus it is a good practice to use the pre-computed kernel technique for optimizing $\mathbf{w}$ and b. In fact, although our method is more

complex, it only takes around 5 minutes for training one model, compared to 40 minutes that was reported in [39].

5.4 Experiment

5.4.1 Dataset

To evaluate our proposed method, we conducted experiments on the large scale TRECVID MED 2012 dataset[3]. This dataset provides the definition for 25 complex events. The first ten event names are listed in Table 5.1. We follow the setting by [39] to divide this video collection into training and testing parts. These parts contain 3,878 and 1,938 videos respectively.

5.4.2 Experimental Setup

At first, original videos are scaled down to 320 x 240 with keeping the aspect ratio. Key frames are sampled at every 2 seconds from the resized video. The segment length is set to 8 seconds as suggested in [86]. To extract feature for each segment, we use the Improved Dense Trajectories feature proposed by Wang and Schmid [90]. Motion Boundary Histogram (MBH) is used to represent extracted trajectories because it can handle camera motion, which is prevalent in realistic videos. For learning, we use our framework jointly with the linear SVM. The cost parameters C_f and C_p are selected by cross-validation in the range of $\{0.1, 1, 10, 100\}$. At the testing step, video-level score is obtained by averaging over all instance scores. Finally we use the standard evaluation metric on MED, Mean Average Precision (mAP), to report the performance.

5.4.3 Baseline Methods

We compare our methods with following baselines: miSVM, MISVM [2], VideoBOW and pSVM [39]. At first, because our method is based on the Multiple Instance Learning (MIL)

[3] http://www.nist.gov/itl/iad/mig/med12.cfm

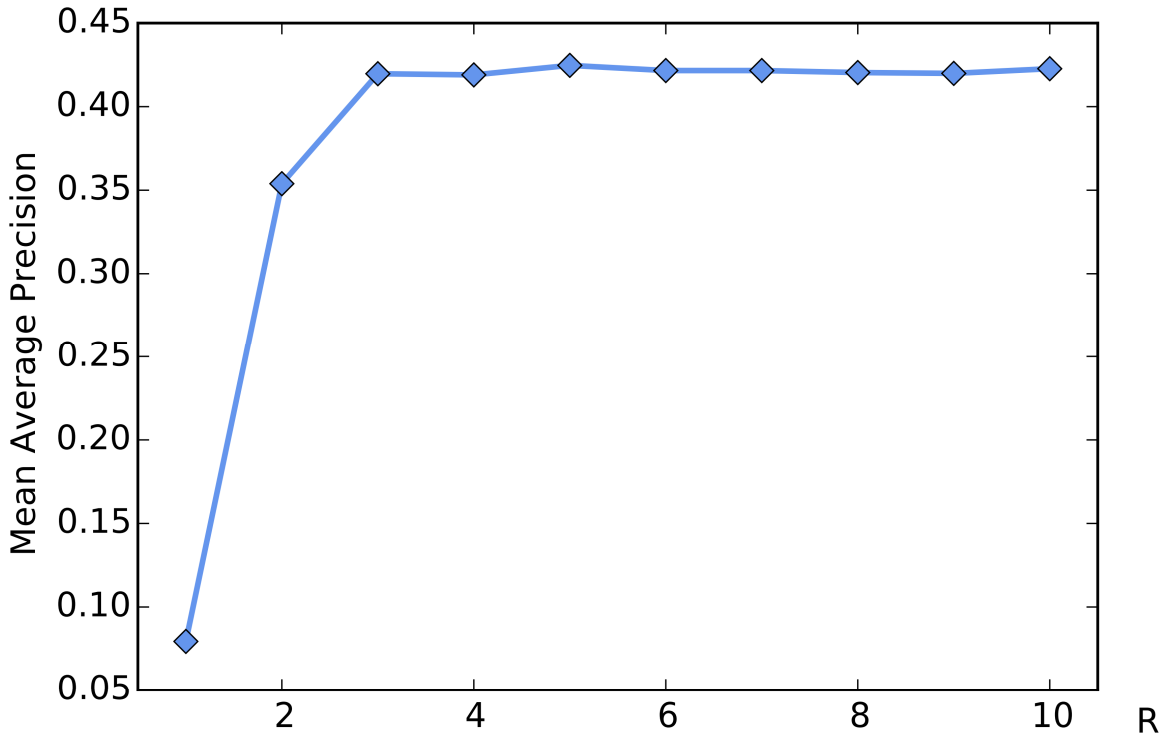


Fig. 5.4 Optimal number of related levels.

framework, we evaluate two MIL solutions: miSVM and MISVM that were proposed by the authors in [2]. The VideoBOW method is the standard approach where local features are aggregated from the whole video. We also compare our method with the recently proposed pSVM which was adopted in [39] for TRECVID MED. For all the baseline methods, except VideoBOW, we utilize the codes provided by the authors to test with our features.

5.4.4 Experimental Results

At first, we conduct experiments to find the optimal value of R. We select R in the range from 1 to 10. The overall performance is shown in Fig. 5.4. We obtain the peak performance with R around 5. Small values of R tend to get low performances. This indicates that the prediction of prior knowledge is not always good, and learning jointly with instance features is necessary. The performance becomes saturated when $R > 5$. Therefore, we fix the value of R to 5 for further experiments.

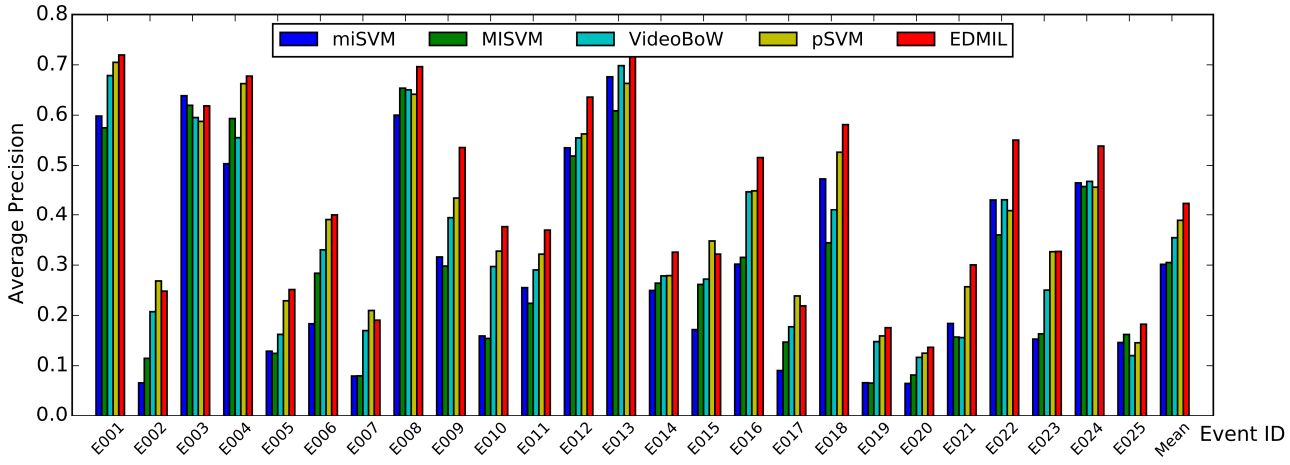


Fig. 5.5 Evaluation results of 25 events in the TRECVID MED 2012 dataset. The mean APs are 0.3015 (miSVM), 0.3051 (MISVM), 0.3544 (VideoBOW), 0.3890 (pSVM) and 0.4246 (Ours).

On The MED 2012 dataset

The performance of each baseline method as well as our method (EDMIL) are shown in Fig. 5.5. Our method significantly outperforms other baselines. For the best baseline, our method relatively outperforms by 10%. Our instance-based classifier can also provide key evidences for event detection. Example of true positive and false positive key evidences detected by our system can be seen in Fig. 5.9 and Fig. 5.10 respectively.

On The MED 2011 dataset

For the MED 2011 dataset, we also compare our proposed method with our two previous works: Segment-based Representation (SB) and Sum-Max Video Pooling (SM) at segment length of 8 s. The results are shown in Fig. 5.6.

Our proposed EDMIL approach achieves the best performance while p-SVM only has a comparable performance with the VideoBOW. Our segment-based approach (SB) does not perform well. The reason is that we used the average aggregation over all segments of the video at the testing step. We further conduct experiment with a new testing strategy: choose the max segment score as the video score. The result of this experiment is shown on Fig. 5.7.

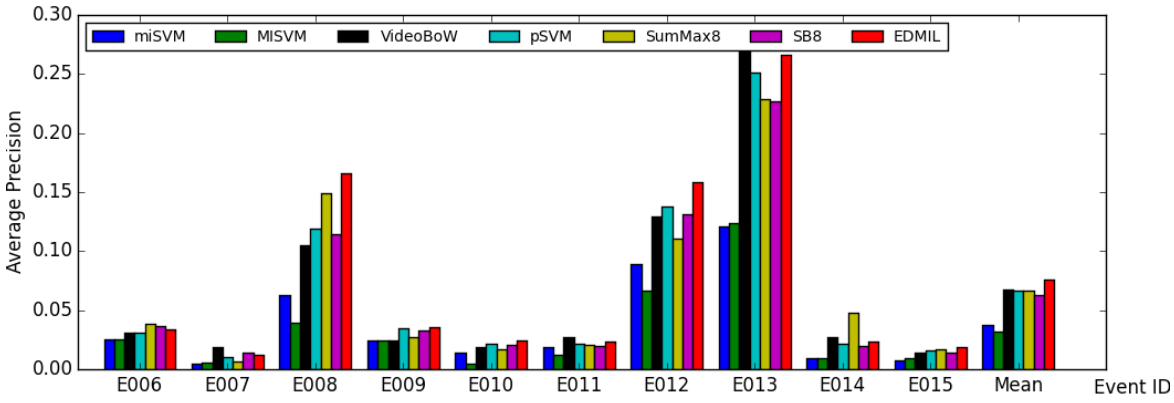


Fig. 5.6 Evaluation results of 10 events in the TRECVID MED 2011 dataset using average aggregation. The mean APs are 0.0378 (miSVM), 0.0322 (MISVM), 0.0674 (VideoBOW), 0.0666 (pSVM), 0.0663 (SM8), 0.0630 (SB8), **0.0761** (**Ours**).

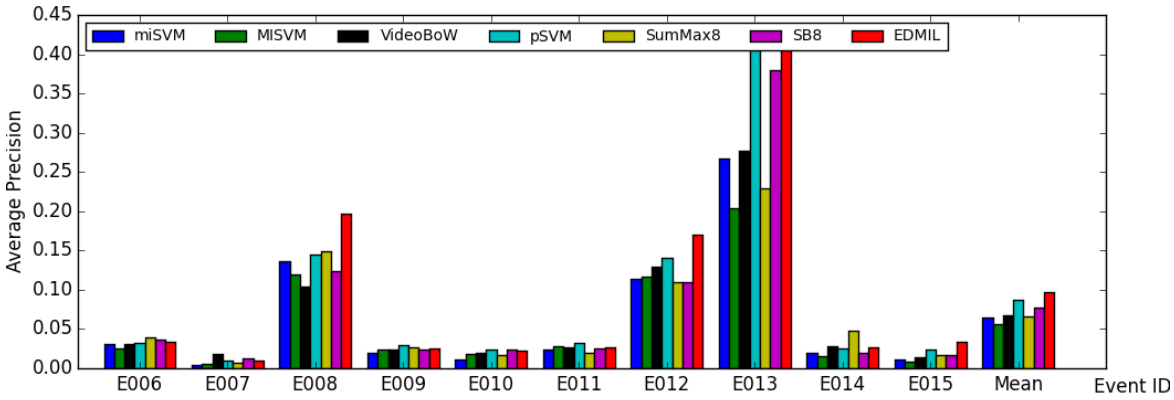


Fig. 5.7 Evaluation results of 10 events in the TRECVID MED 2011 dataset using max aggregation. The mean APs are 0.0640 (miSVM), 0.0564 (MISVM), 0.0674 (VideoBOW), 0.0870 (pSVM), 0.0663 (SM8), 0.0770 (SB8), **0.0968** (**Ours**).

The max aggregation strategy performs better than the average aggregation by a margin. The performance gain can be seen in Fig. 5.8.

5.5 Conclusion

We propose a new method to detect event in videos from its key evidences. Our method differs from others in that we utilize the evidential description provided for each event. Given this supportive information, we search for key evidences by jointly optimizing with instance

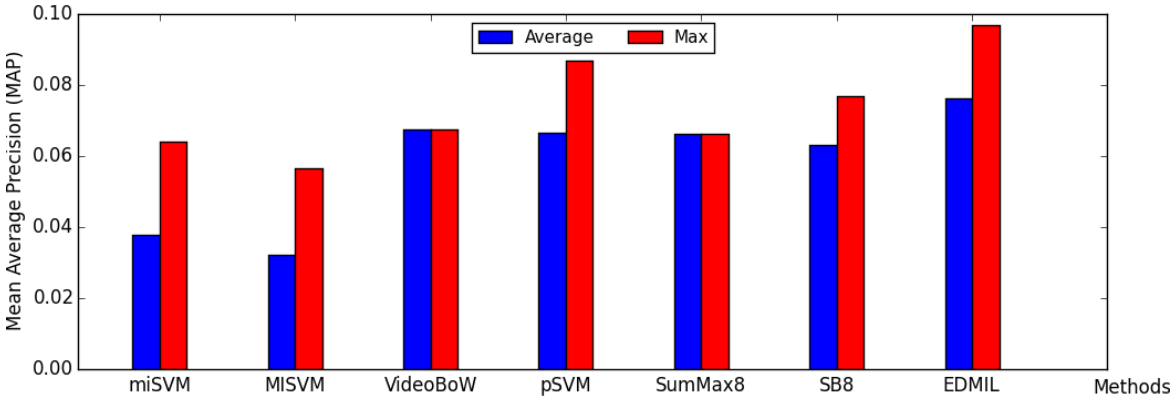


Fig. 5.8 The top 6 key evidences detected by our system for the event "Attempting board trick". The dominance of ski-related instances is reasonable.

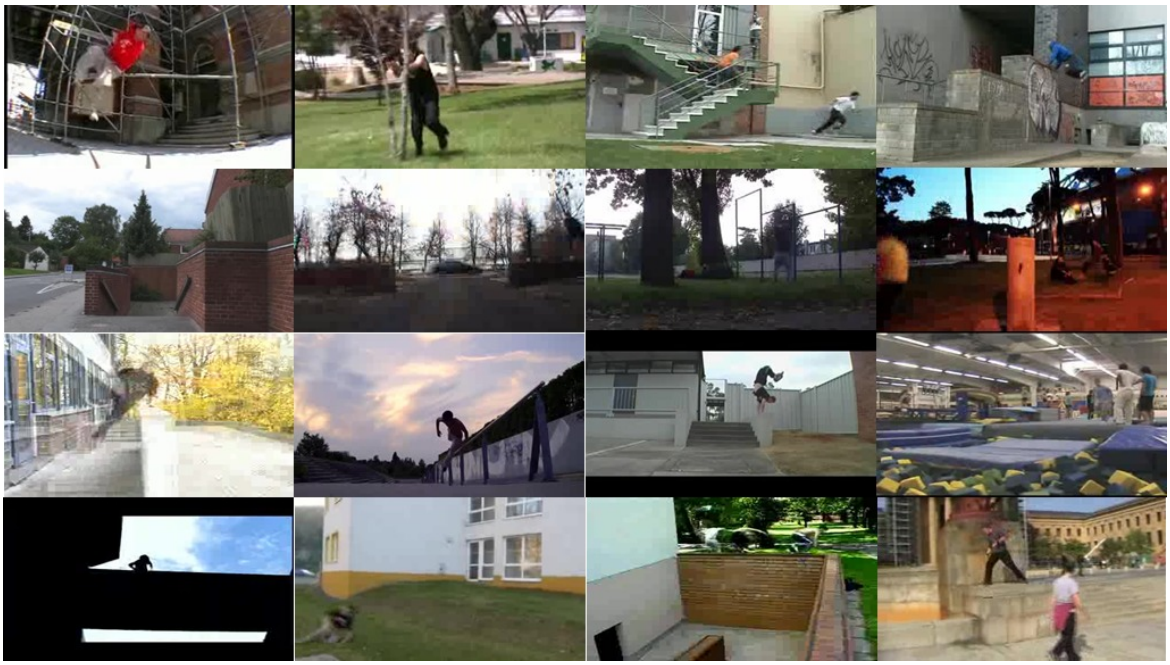


Fig. 5.9 The top 16 key evidences detected by our system for the event "Parkour".

Fig. 5.10 The top 16 key false positive evidences detected by our system for the event "Parkour".

feature in a variant of multiple instance learning framework. As a result, we obtained a superior event detection performance.

Chapter 6

Conclusion

If you can't fly then run, if you can't run then walk, if you can't walk then crawl, but whatever you do you have to keep moving forward.

– Martin Luther King Jr.

6.1 Summary

Recognizing complex event in videos has become an important task in computer vision due to various applications. However, this is a challenging task because we have to deal with real videos. In summary, there are four main challenges that we need to handle:

1. *Large content variation*.
2. *Uncontrolled capturing condition.*
3. *Large scale video dataset.*
4. *Near-miss videos.*

The most important challenge that need to be handled is *uncontrolled capturing condition*. This challenge of internet videos often harm the performance of event detection systems that was built on action recognition techniques. We handle this challenge by decomposing the

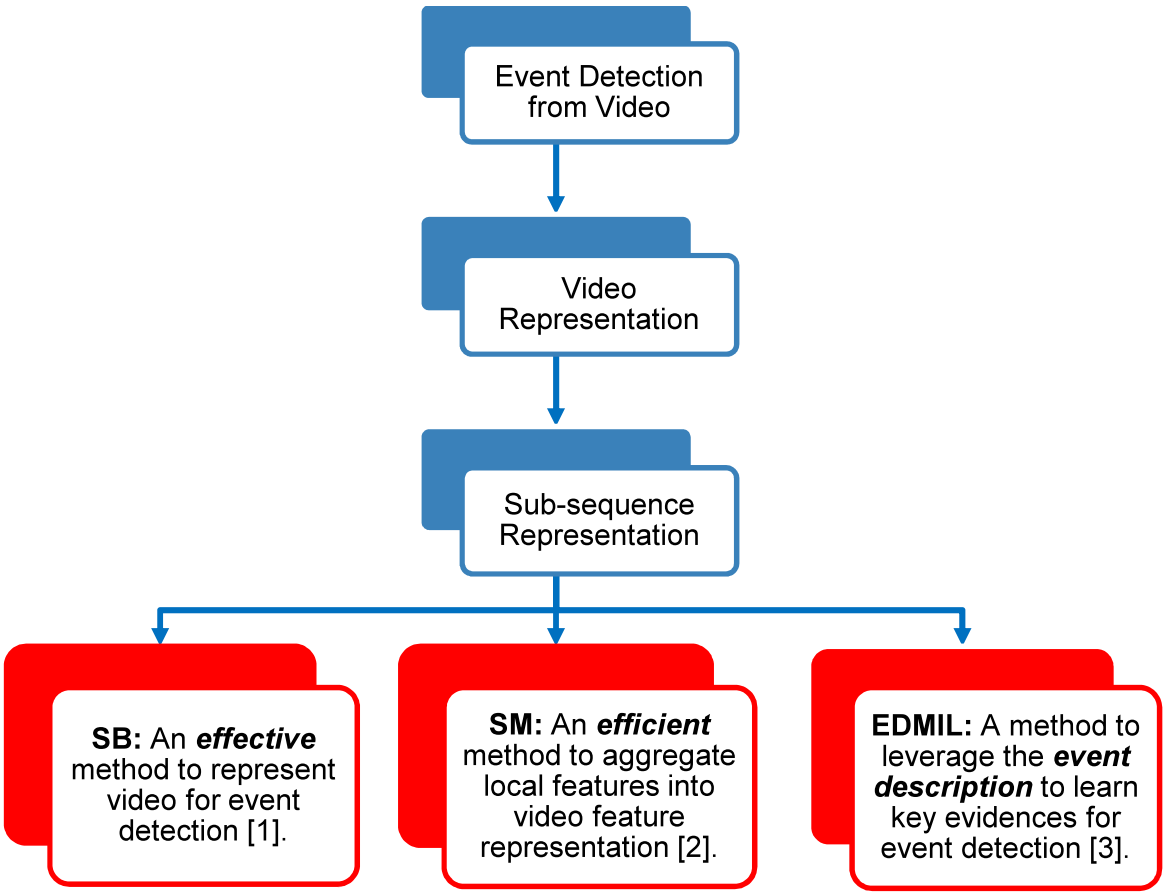


Fig. 6.1 Summary of contributions of my dissertation.

original videos into segments and investigating *feature representation*, *feature aggregation*, and *feature learning* methods from these segments. Beside this main challenge, we also deal with the *large content variation* and *large scale video dataset* as well. To this end, we made following contributions (Fig. 6.1):

1. We propose a new *feature representation* method, named segment-based representation (**SB**), to overcome the limitations of the traditional video-based approaches. The basic idea is to examine shorter segments instead of using the representative frames or entire video. We carry thorough experiments to verify our proposed method by investigating different strategies to decompose a video into segments. These strategies include uniform segment sampling and segments based on shot boundary detection. By using

more training examples (at segment level), this method can handle the *large content variation* challenge as well.

2. We propose a new *feature aggregation* method, called sum-max video pooling (**SM**), to deal with noisy information in complex videos. This pooling technique is based on the layer structure of video. Basically, we apply sum pooling at the low layer representation while using max pooling at the high layer representation. Sum pooling is used to keep sufficient relevant features at the low layer, while max pooling is used to retrieve the most relevant features at the high layer, therefore it can discard irrelevant features in the final video representation. Our video pooling method is very efficient, thus it can be applied to *large scale video dataset* as well.

3. We propose a new *feature learning* method, named Event-driven Multiple Instance Learning (**EDMIL**), to learn key evidences for complex event detection. We treat each segment as an instance and model it in a multiple instance learning framework [2], where each video is a "bag". The instance-event similarity is quantized into different levels of relatedness. Intuitively, the most (ir)relevant instances should have higher (dis)similarities. Therefore, we propose to learn the instance labels by jointly optimizing the instance classifier and its related level. Similar to the first contribution, this method also use more training examples (at segment level), therefore it can handle the *large content variation* challenge.

It is beneficial to use some engineering tricks in order to handle large scale dataset. For example, the pre-computed kernel is suitable when there is a large number of events. In this case, we only need to calculate the kernel one time and train multiple time with different labels. This technique is especially useful in our EDMIL method.

A summary of the significant achievement of our proposed methods can be seen in Fig. 6.2. Our methods (SB, SM and EDMIL) can improve the baseline VideoBOW by **22.55%**, **2.67%** and **43.62%** respectively on the large scale MED 2011 dataset.

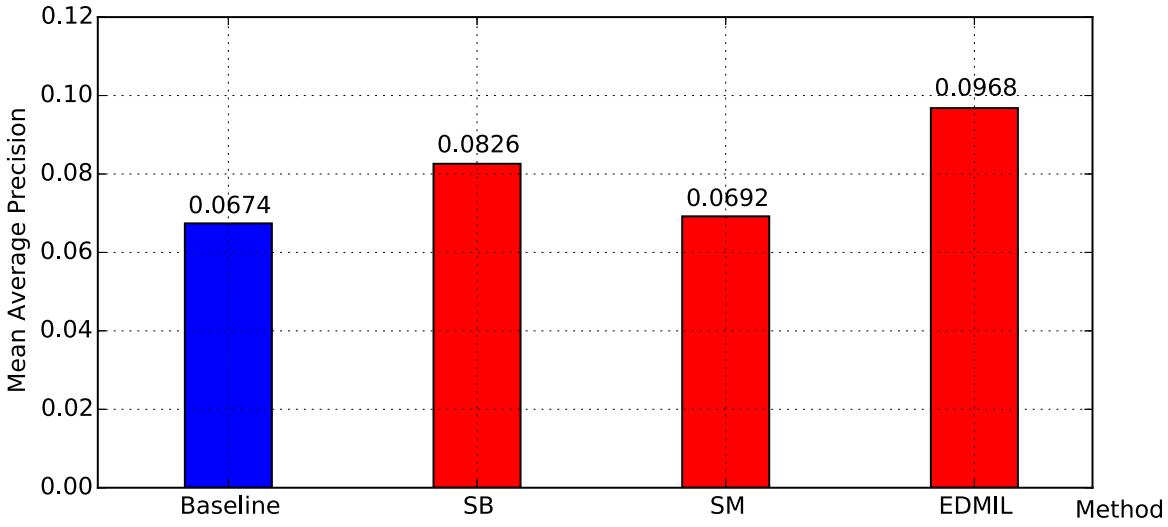


Fig. 6.2 Performance comparison of our proposed solutions on the large scale MED 2011 dataset.

6.2 Conclusion

Detecting event in video is a challenging yet worth pursuing research topic. Due to nature of internet videos, which often contains irrelevant content, it is crucial to develop robust technologies for event detection in realistic videos. This dissertation has addressed this challenging issue by introducing three major techniques including a feature representation, feature aggregation and feature learning method.

At first, we proposed using the segment-based approach for event detection. Our proposed segment-based approach outperforms the video-based approach in most cases when using a simple non-overlapping sampling strategy. More interestingly, the results are significantly improved when we using the segment-based approach with an overlapping sampling strategy. This suggests the importance of segment localization on the event detection performance. Suppose the segment length is fixed, we are interested in determining which segment is the best representative for an event. In this study, we also observed that the detection performance is quite sensitive to the segment-length and it depends on the dataset. The results obtained from the late fusion strategy is quite stable and close the peak performance. This suggests a methodical way to generalize the segment-based approach to other datasets. However, this

method is not scalable because it requires a lot of computation costs. Therefore, learning an optimal segment length for each event can be beneficial for an event detection system.

Secondly, we proposed to use a sum-max video pooling technique to combine both sum pooling and max pooling into a holistic video representation. This pooling technique is based on the layered structure of video. Preliminary results showed that this is an promising direction for video representation. One limitation of the current approach is that the performance depends on the segment length. Therefore, we suggest to investigate a better approach to utilize the layered structure of video for video representation.

Lastly, we proposed a new feature learning method to detect event in videos from its key evidences. Our method differs from others in that we utilize the evidential description provided for each event. Given this supportive information, we search for key evidences by jointly optimizing with instance feature in a variant of multiple instance learning framework. As a result, we obtained a superior event detection performance.

6.3 Future Work

We plan to extend our work in following directions.

- **Learning the relationship between segments**. Currently, we can learn a set of important segments that can be used for event detection. We have not imposed any constraints on the relation between segments. However, some spatial-temporal relationship might be important to identify an event. For example, in the event "changing a vehicle tire", the action "removing hubcap" should take place before the action "replacing tire". Or in the event "flash mob gathering", the "gathering" action should happen before the "dancing" action takes place. Moreover, some actions can have a co-occurrence relationship. For example, in the "birthday party" event, people can be both singing and dancing.

- **Learning the importance of each concept in the concept bank** for event detection. Currently we only detect a set of concepts that can be used to provide evidences to detect an event. These concepts are obtained from NLP techniques. However, we do

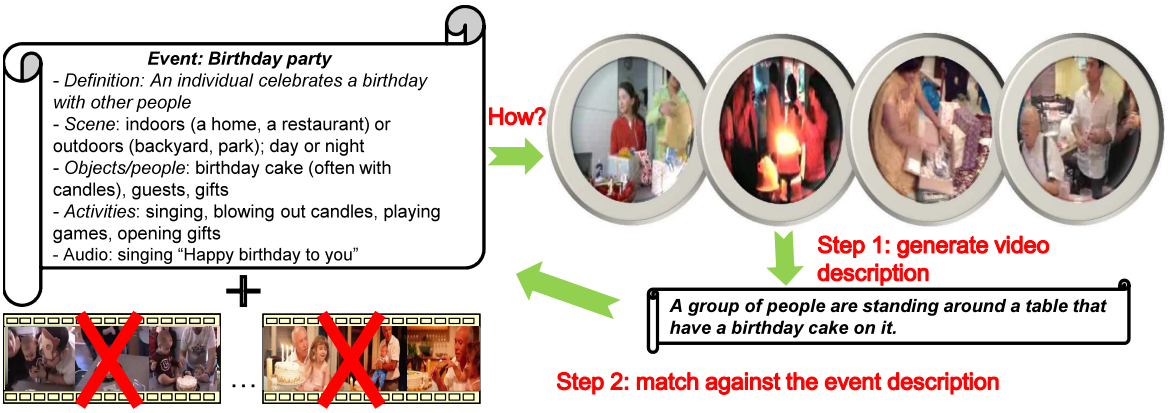


Fig. 6.3 Illustration of video event description and video event detection.

not know if it really visually represents for that event. It is interesting know which concepts that both textually and visually represent for an event.

- **Video description generation**. This is the task that describing about what happening in a video. This task also has many practical applications such as helping blind or visually impaired people understand what happening in videos. Besides, it can be used to build question-answering systems, which provides an interactive mechanism for a better understanding of the video. Moreover, this technology, as a result, can be applied to zero-shot event detection, as illustrated in Fig. 6.3.

References

[1] Aly, R., Arandjelovic, R., Chatfield, K., Douze, M., Fernando, B., Harchaoui, Z., McGuinness, K., O'Connor, N. E., Oneata, D., Parkhi, O. M., et al. (2013). The axes submissions at trecvid 2013.

[2] Andrews, S., Tsochantaridis, I., and Hofmann, T. (2002). Support vector machines for multiple-instance learning. In *NIPS*, pages 561–568.

[3] Arandjelovic, R. and Zisserman, A. (2012). Three things everyone should know to improve object retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2911–2918. IEEE.

[4] Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., et al. (2007). Greedy layer-wise training of deep networks.

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

[6] Brookes, M. (2003). Voicebox: Speech processing toolbox for matlab.

[7] Burghouts, G. J. and Geusebroek, J.-M. (2009). Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, 113(1):48–62.

[8] Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

[9] Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

[10] Chen, J., Cui, Y., Ye, G., Liu, D., and Chang, S.-F. (2014). Event-driven semantic concept discovery by exploiting weakly tagged internet images. In *ICMR*.

[11] Chen, M. and Hauptmann, A. (2009). Mosift: Recognizing human actions in surveillance videos. In *Computer Science Department, CMU-CS-09-161*.

[12] Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

[13] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334.

[14] Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *European Conference on Computer Vision*. Springer.

[15] Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B., et al. (2010). The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296. ACM.

[16] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

[17] Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *VS-PETS*.

[18] Duan, L., Xu, D., and Chang, S.-F. (2012a). Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1338–1345. IEEE.

[19] Duan, L., Xu, D., Tsang, I. W.-H., and Luo, J. (2012b). Visual event recognition in videos by learning from web data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1667–1680.

[20] Ellis, D. P. W. (2005). PLP and RASTA (and MFCC, and inversion) in Matlab. online web resource.

[21] F. Yuan, Gui-Song Xiab, H. S. V. P. (2012). Spatio-temporal context of mid-level features for activity recognition. In *Pattern Regonition Letters 2012 (submitted)*.

[22] Guimarães, S. J. F., Couprie, M., Araújo, A. d. A., and Leite, N. J. (2003). Video segmentation based on 2d image analysis. *Pattern Recogn. Lett.*, 24(7):947–957.

[23] Habibian, A., van de Sande, K. E., and Snoek, C. G. (2013). Recommendations for video event recognition using concept vocabularies. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 89–96. ACM.

[24] Harris, Z. S. (1954). Distributional structure. *Word*.

[25] Hill, M., Hua, G., Natsev, A., Smith, J. R., Xie, L., Huang, B., Merler, M., Ouyang, H., and Zhou, M. (2010). Ibm research trecvid-2010 video copy detection and multimedia event detection system. In *NIST TRECVID Workshop*, Gaithersburg, MD.

[26] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

[27] internetworldstats.com (2014). Internet users in the world distribution by world regions - 2014 q2.

[28] Jaakkola, T., Haussler, D., et al. (1999). Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493.

[29] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM.

[30] Jiang, Y., Yuan, J., and Yu, G. (2012). Randomized spatial partition for scene recognition. In *ECCV (2)*, pages 730–743.

[31] Jiang, Y.-G., Ngo, C.-W., and Yang, J. (2007). Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and Video Retrieval*, pages 494–501.

[32] Jiang, Y.-G., Yang, J., Ngo, C.-W., and Hauptmann, A. G. (2010a). Representations of keypoint-based semantic concept detection: A comprehensive study. *Multimedia, IEEE Transactions on*, 12(1):42–53.

[33] Jiang, Y.-G., Zeng, X., Ye, G., Bhattacharya, S., Ellis, D., Shah, M., and Chang, S.-F. (2010b). Columbia-ucf trecvid2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching. In *NIST TRECVID Workshop*, Gaithersburg, MD.

[34] Jiang, Y.-G., Zeng, X., Ye, G., Ellis, D., Chang, S.-F., Bhattacharya, S., and Shah, M. (2010c). Columbia-ucf trecvid2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching. In *TRECVID*.

[35] Kläser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, pages 995–1004.

[36] Koenderink, J. J. and Van Doorn, A. J. (1999). The structure of locally orderless images. *Int. J. Comput. Vision*, 31(2-3):159–168.

[37] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[38] Lai, K.-T., Liu, D., Chen, M.-S., and Chang, S.-F. (2014a). Recognizing complex events in videos by learning key static-dynamic evidences. In *ECCV*.

[39] Lai, K.-T., Yu, F. X., Chen, M.-S., and Chang, S.-F. (2014b). Video event detection by inferring temporal instance labels. In *CVPR*, pages 2251–2258. IEEE.

[40] Lan, Z.-z., Bao, L., Yu, S.-I., Liu, W., and Hauptmann, A. G. (2012). *Double fusion for multimedia event detection*. Springer.

[41] Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123.

[42] Laptev, I. and Lindeberg, T. (2003). Space-time interest points. In *ICCV*, pages 432–439.

[43] Laptev, I., Marszałek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Conference on Computer Vision & Pattern Recognition*.

[44] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE.

[45] LeCun, Y. and Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10).

[46] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[47] Lee, C.-H., Soong, F., and Juang, B.-H. (1988). A segment model based approach to speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing, 1988. ICASSP-88.*, pages 501 –541 vol.1.

[48] Liu, C., Yuen, J., and Torralba, A. (2009a). Nonparametric scene parsing: Label transfer via dense scene alignment. In *Computer Vision and Pattern Recognition*.

[49] Liu, J., Luo, J., and Shah, M. (2009b). Recognizing realistic actions from videos “in the wild”. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1996–2003. IEEE.

[50] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[51] Marszałek, M., Laptev, I., and Schmid, C. (2009). Actions in context. In *IEEE Conference on Computer Vision & Pattern Recognition*.

[52] Mathieu, B., Essid, S., Fillon, T., Prado, J., and Richard, G. (2010). Yaafe, an easy to use and efficient audio feature extraction software.

[53] Matikainen, P., Hebert, M., and Sukthankar, R. (2009). Trajectons: Action recognition through the motion analysis of tracked features. In *Workshop on Video-Oriented Object and Event Classification, ICCV 2009*.

[54] Matsuo, T. and Nakajima, S. (2010). Nikon multimedia event detection system. In *NIST TRECVID Workshop*, Gaithersburg, MD.

[55] Mei, T., Hua, X.-S., Yang, B., Yang, L., and Li, S. (2007). Automatic video recommendation. US Patent App. 11/771,219.

[56] Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In *Computer Vision—ECCV 2002*, pages 128–142. Springer.

[57] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630.

[58] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[59] Myers, G. K., Nallapati, R., van Hout, J., Pancoast, S., Nevatia, R., Sun, C., Habibian, A., Koelma, D. C., van de Sande, K. E., Smeulders, A. W., et al. (2014). Evaluating multimedia features and fusion for example-based event detection. *Machine Vision and Applications*, 25(1):17–32.

[60] Natarajan, P., Manohar, V., Wu, S., Tsakalidis, S., Vitaladevuni, S. N., Zhuang, X., Prasad, R., Ye, G., and Liu, D. (2011). Bbn viser trecvid 2011 multimedia event detection system. In *NIST TRECVID Workshop*, Gaithersburg, MD.

[61] Natarajan, P., Natarajan, P., Wu, S., Zhuang, X., Vazquez-Reina, A., Vitaladevuni, S. N., Tsourides, K., Andersen, C., Prasad, R., Ye, G., Liu, D., Chang, S., Saleemi, I., Shah, M., Ng, Y., White, B., Gupta, A., and Haritaoglu, I. (2012a). Bbn viser trecvid 2012 multimedia event detection and multimedia event recounting systems. In *NIST TRECVID Workshop*, Gaithersburg, États-Unis.

[62] Natarajan, P., Wu, S., Vitaladevuni, S., Zhuang, X., Tsakalidis, S., Park, U., Prasad, R., and Natarajan, P. (2012b). Multimodal feature fusion for robust event detection in web videos. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1298–1305. IEEE.

[63] News, B. (2015). Facebook restricts violent video clips and photos.

[64] Nowak, E., Jurie, F., and Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *Computer Vision–ECCV 2006*, pages 490–503. Springer.

[65] Oh, S., McCloskey, S., Kim, I., Vahdat, A., Cannons, K. J., Hajimirsadeghi, H., Mori, G., Perera, A. A., Pandey, M., and Corso, J. J. (2014). Multimedia event detection with multimodal feature fusion and temporal concept localization. *Machine vision and applications*, 25(1):49–69.

[66] Oliva, A. and Torralba, A. (2001a). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.

[67] Oliva, A. and Torralba, A. (2001b). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175.

[68] Oneata, D., Douze, M., Revaud, J., Jochen, S., Potapov, D., Wang, H., Harchaoui, Z., Verbeek, J., Schmid, C., Aly, R., Mcguiness, K., Chen, S., O'Connor, N., Chatfield,

K., Parkhi, O., Arandjelovic, R., Zisserman, A., Basura, F., and Tuytelaars, T. (2012). AXES at TRECVid 2012: KIS, INS, and MED. In *TRECVID Workshop*, Gaithersburg, États-Unis.

[69] Oneata, D., Verbeek, J., and Schmid, C. (2013). Action and event recognition with fisher vectors on a compact feature set. In *ICCV*. IEEE.

[70] Oneata, D., Verbeek, J., and Schmid, C. (2014). The lear submission at thumos 2014.

[71] Over, P., Awad, G. M., Fiscus, J., Antonishek, B., Michel, M., Smeaton, A. F., Kraaij, W., and Quénot, G. (2011). Trecvid 2010–an overview of the goals, tasks, data, evaluation mechanisms, and metrics.

[72] Over, P., Fiscus, J., Sanders, G., Joy, D., Michel, M., Awad, G., Smeaton, A., Kraaij, W., and Quénot, G. (2014). Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID 2014*. National Institute of Standards and Technology.

[73] Perronnin, F., Sánchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer.

[74] Phan, S., Ngo, T. D., Lam, V., Tran, S., Le, D.-D., Duong, D. A., and Satoh, S. (2014). Multimedia event detection using segment-based approach for motion feature. *Signal Processing Systems*, 74(1):19–31.

[75] Rabiner, L. R. and Schafer, R. W. (2007). Introduction to digital speech processing. *Foundations and trends in signal processing*, 1(1):1–194.

[76] Rodriguez, M., Ahmed, J., and Shah, M. (2008). Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

[77] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, DTIC Document.

[78] Sánchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245.

[79] Serre, T., Wolf, L., and Poggio, T. (2005). Object recognition with features inspired by visual cortex. pages 994–1000.

[80] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[81] Sivic, J. and Zisserman, A. (2009). Efficient visual search of videos cast as text retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):591–606.

[82] Snoek, C. G., Worring, M., and Smeulders, A. W. (2005). Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM.

[83] Sun, C. and Nevatia, R. (2013). Large-scale web video event classification by use of fisher vectors. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 15–22. IEEE.

[84] Tang, K., Fei-Fei, L., and Koller, D. (2012). Learning latent temporal structure for complex event detection. In *CVPR*, pages 1250–1257. IEEE.

[85] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2014). C3d: generic features for video analysis. *arXiv preprint arXiv:1412.0767*.

[86] Vahdat, A., Cannons, K., Mori, G., Oh, S., and Kim, I. (2013). Compositional models for video event detection: A multiple kernel learning latent variable approach. In *ICCV*, pages 1185–1192. IEEE.

[87] van de Sande, K. E. A., Gevers, T., and Snoek, C. G. M. (2010). Evaluating color descriptors for object and scene recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 32, pages 1582–1596.

[88] Vedaldi, A. and Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms.

[89] Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2011). Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, Colorado Springs, United States.

[90] Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3551–3558. IEEE.

[91] Willems, G., Tuytelaars, T., and Van Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *Computer Vision–ECCV 2008*, pages 650–663. Springer.

[92] Wu, S., Bondugula, S., Luisier, F., Zhuang, X., and Natarajan, P. (2014). Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *CVPR*, pages 2665–2672. IEEE.

[93] Xu, D. and Chang, S.-F. (2008). Video event recognition using kernel methods with multilevel temporal alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1985–1997.

[94] Xu, Z., Yang, Y., Tsang, I., Sebe, N., and Hauptmann, A. G. (2013). Feature weighting via optimal thresholding for video analysis. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3440–3447. IEEE.

[95] Yu, F., Liu, D., Kumar, S., Tony, J., and Chang, S.-F. (2013). psvm for learning with label proportions. In *Proceedings of The 30th International Conference on Machine Learning*, pages 504–512.

[96] Yu, S.-I., Jiang, L., Mao, Z., Chang, X., Du, X., Gan, C., Lan, Z., Xu, Z., Li, X., Cai, Y., et al. (2014). Informedia@ trecvid 2014 med and mer. In *NIST TRECVID Video Retrieval Evaluation Workshop*.

[97] Yuan, J., Liu, Z., and Wu, Y. (2011). Discriminative video pattern search for efficient action detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1728 –1743.

[98] Zhang, H. J., Wu, J., Zhong, D., and Smoliar, S. W. (1997). An integrated system for content-based video retrieval and browsing. *Pattern recognition*, 30(4):643–658.

[99] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495.

[100] Zillmann, D. and Weaver, J. B. (1999). Effects of prolonged exposure to gratuitous media violence on provoked and unprovoked hostile behavior1. *Journal of Applied Social Psychology*, 29(1):145–165.

Appendix A

TRECVID MED 2013 Results

In this appendix, we briefly introduce our Multimedia Event Detection system for TRECVID MED 2013. We use both audio and visual features with Bag-of-Words and Fisher Vector Representation. Our MED framework consists of following steps: preprocessing, feature extraction, feature representation and event classification.

A.1 Preprocessing

At first, all videos are normalized to around 320x240. We fix the width dimension to 320 and change the height so that the aspect ratios are kept. The audio channels are removed from resized videos to save disk space. After that, we extract one representative keyframe from resized videos at every 2 seconds and audio feature from the original videos.

A.2 Feature Extraction

We use feature from different modalities to model multimedia events: still image features, motion features and audio features. We use the standard SIFT with Hessian Laplace detector for extracting still image feature. For motion feature, we use Dense Trajectories with MBH descriptor. We use the MFCC for extracting audio feature.

A.3 Feature Representation

Bag-of-Words representation is a simple way to encode local features. It is the frequency histogram of local descriptors that are assigned to the nearest clusters. In the implementation, we randomly select 1,000,000 local descriptors to train the codebook with 4,000 codewords. The soft assignment technique is also employed to reduce the quantization errors. For Fisher vector, we use the codebook size of 256 clusters which are generated using the Gaussian Mixture Model (GMM). We further improve the expressiveness of Fisher vector by applying PCA for reducing feature dimension, i.e 80-d for SIFT and 128-d for MBH.

A.4 Event Classification

We use the popular Support Vector Machine (SVM) for classification. All the positive videos are considered as positive samples and the remaining videos are considered as negative samples (including near miss videos). We use the chi-square kernel for training bag-of-words histogram features and linear kernel for training features encoded by Fisher vector.

A.5 Result and Conclusion

We observed that Fisher vector representation is consistently better the traditional bag-of-words histogram representation. The motion features archived the highest performance in terms of single feature comparison, followed by image features and audio features. Furthermore, these features are highly complementary, so their combination achieved the best performance. We also observed a little performance gain when combining both Fisher vector and bag-of-words feature encoding. Based on these observations, we submitted the FullSys system based on the combination of audio or visual features. Our results (NII Team) on the 100Ex setting is shown in Fig. A.1. Our rank is 4th out of 18 participants.

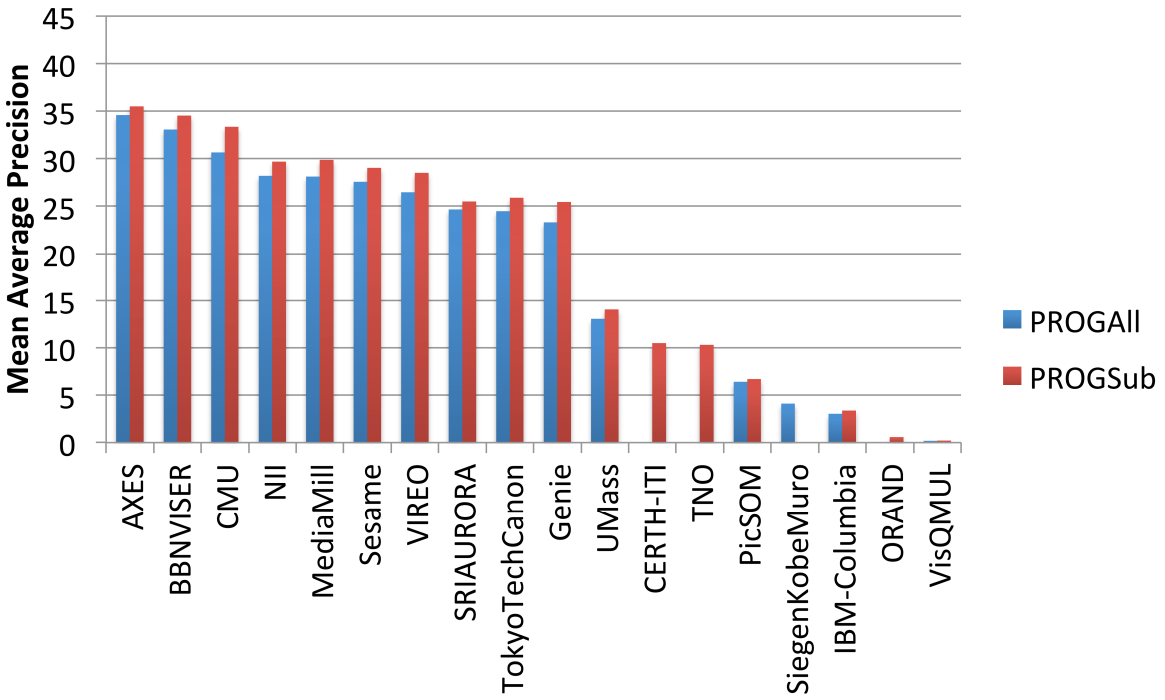


Fig. A.1 Comparison of our MED 2013 system with others on the full evaluation set for the Pre-specified task. Results are sorted in the descending order of performance on the EK100 setting.

Appendix B

TRECVID MED 2014 Results

In MED 2014, we study some technical improvements for motion feature and image features over our MED 2014 System.

B.1 For Motion Feature

We use the improved version of Dense Trajectories motion feature [90]. To describe trajectories, we choose to use both HOGHOF and MBH descriptors, which have been proved to be effective for MED by AXES team [1]. In order to combine these descriptors, we train two independent GMM codebooks. After that Fisher vector is used to encode feature from each descriptor independently. The resulting representation at video level of each descriptor is normalized by power normalization and L2 normalization. Finally these two feature vectors are concatenated to form the final representation of each video.

B.2 For Image Feature

We apply two technical improvements on the image feature. At first, a new way of video level feature representation is used to pool feature from its keyframe-based representation. In MED 2013 system, we aggregated local descriptors from all sampled frames in video without explicitly calculating keyframe-based features. For this year's system, Fisher vector

Table B.1 Performance comparison of different motion feature configurations.

MED13 System	MED14 System	
Dense Trajectories (MBH)	Improved Dense Trajectories (MBH)	Improved Dense Trajectories (HOGHOF + MBH)
28.33	35.07	40.77

Table B.2 Performance comparison of different image feature configurations.

MED13 System	MED14 System	
SIFT	SIFT (New aggregation)	SIFT (New aggregation + RootSIFT)
23.41	24.24	27.02

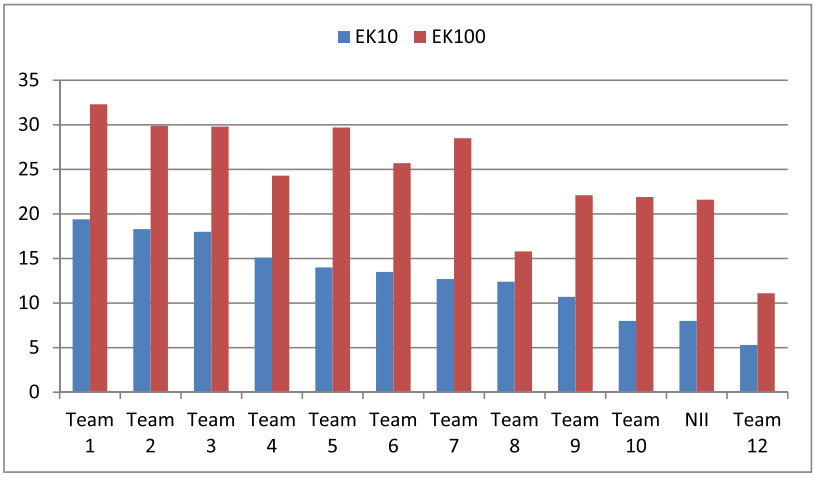
is encoded for each sampled frame and normalized using power and L2 normalization. Features from these sampled frames are averaged to form the video level representation. The second technical improvement is using RootSIFT features [3]. We have applied RootSIFT with different implementation of SIFT features such as the one use in [57], VLFeat [88], and Color Descriptor [87]. Finally we chose to use VLFeat because it achieved the best performance in our evaluation framework.

We evaluated the performance of new components on the KINDREDTEST 13 dataset. All results are reported in terms of Mean Average Precision (MAP). Performance comparison of motion features and image features are shown in Table B.1 and Table B.2 respectively.

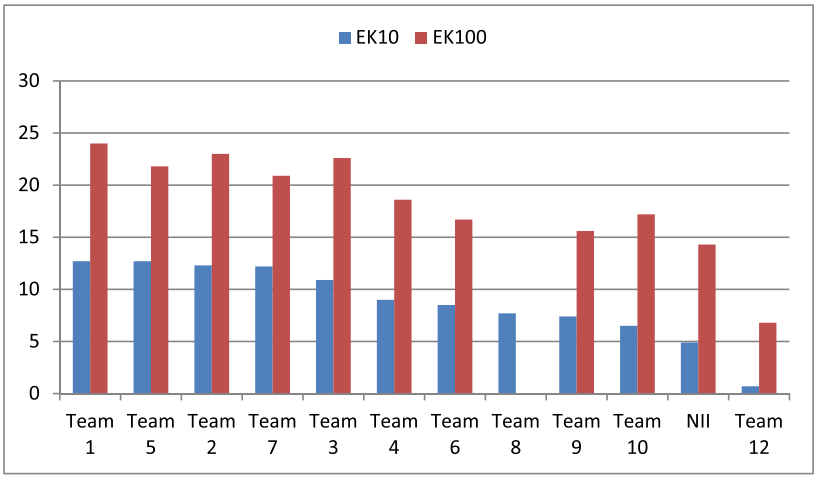
Unfortunately, we could not finish running the best configuration for motion features, so we use the same configuration as previous year because it took less time. For image feature, we used the improved version. We also used the late fusion technique to combine audio and visual features in our final submission. For related videos, we fixed our system to use them as negative training samples for both EK10 and EK100 settings. We participated in the full evaluation set containing around 200K videos for both Pre-specified (PS) and Adhoc (AH) tasks.

B.3 Results and Conclusion

Results of our MED system is shown in Fig. B.1. Our ranks was 11th out of 12 teams in the EK10 setting and 10th in the EK100 setting. This observation is same for both PS and AH tasks. Compared to top MED systems, our system is significantly worse in the EK10 setting. For example, our performance are 67% and 41% relatively to the best MED system in the EK100 and EK10 respectively. We have learnt that top performance system have incorporated semantic concept detection, which can be more helpful when number of training videos are limited. This might be the reason for the significant drop on the performance of our EK10 system.



(a) Pre-Specified systems



(b) Ad-Hoc Systems

Fig. B.1 Comparison of our MED 2014 system with others on the full evaluation set for both Pre-specified and Ad-hoc tasks. Results are sorted in the descending order of performance on the EK10 setting.

Appendix C

TRECVID MED 2015 Results

In TRECVID MED 2015, beside using audio and visual features with Fisher vector encoding, we also use deep learning features extracted from a pre-trained model. The features from the output of the final layer are also employed to zero shot event detection. Our results demonstrated the benefit of using deep learning features, especially in the case of less training examples.

C.1 Improvements over MED'14 System

DCNN features. We use the popular DeepCaffe [29] framework to extract image features. We used the pre-trained deep model provided by Zhou et al. [99]. This model was trained on an image collection of 1,183 categories including 205 scene categories from the Places Database and 978 object categories from the ImageNet 2012. We selected the neuron activations from the last three layers for the feature representation. The third and second-to-last layer has 4,096 dimensions, while the last layer has 1,183 dimensions. We denote these features as FC6, FC7, and FULL in our experiments.

Zero-shot event detection. In order to calculate the similarity between an video and an event, we adopt a concept expansion strategy as in [10]. The outline of our method is illustrated in Fig. 5.3 and it consists of four steps: concept detection, event representation, concept-event similarity and instance-event similarity (as described in detail in Section 5.2).

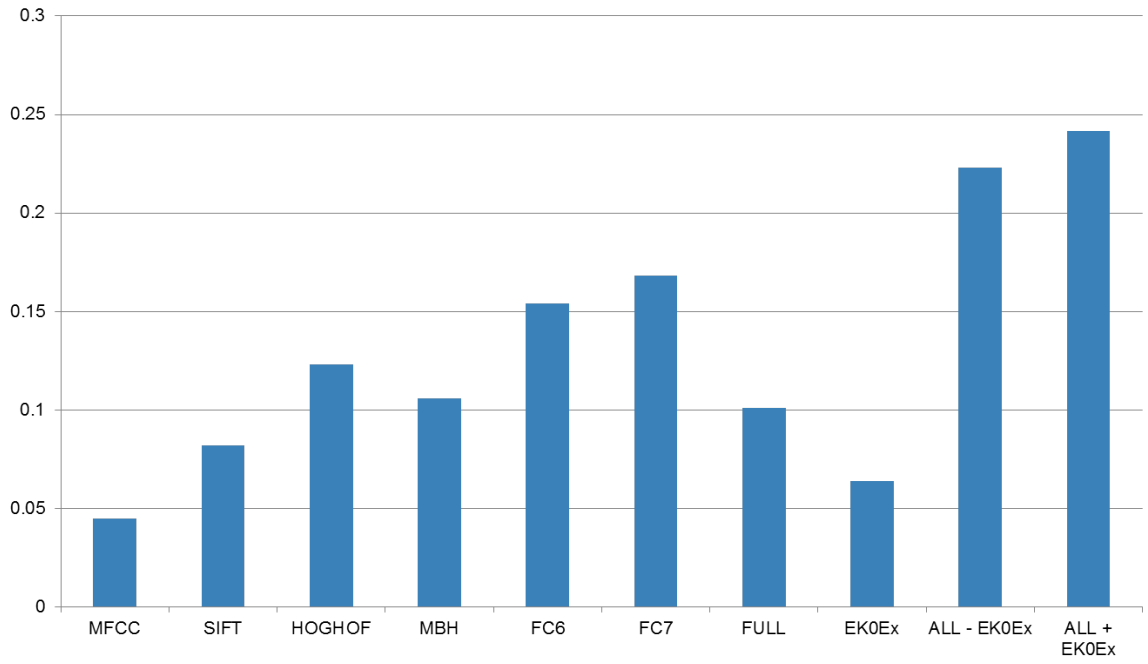


Fig. C.1 Performance of each feature and the fused runs.

C.2 Contribution of New Components

We evaluated the contribution of new components on the KINDREDTEST14 dataset. All results are reported in terms of Mean Average Precision (MAP). Here we only report the over all performance, which is averaged from all events. In figure C.1, we show the performance of each feature, including low level features for comparison.

DCNN Features. In terms of single feature performance, FC7 feature has the best performance, even better than dense trajectories feature. It can be seen that feature of the last layer (FULL) does not perform well. It is due to the lost of information after applying max pooling from the previous layer.

Zero-shot Event Detection. Performance of our EK0 run is around 6% MAP, which is slightly better than the audio MFCC run. Moreover, this run is complementary to low level and deep learning features. Combining with all low level and deep learning features, we obtained around 8% relative improvement, as shown in the last column of Fig. C.1.

C.3 Submitted Systems

After evaluating the improvements on the KINDREDTEST dataset, we chose to submit the run that combining all available features for EK10 and EK100 settings. We also submit our EK0 system to the full evaluation.

C.4 Result and Conclusion

Results of our MED system on the full evaluation set is shown in Fig. C.2. Performance is reported in terms of MAP. Comparing with other systems, we are ranked 6th out of 7 teams in the EK10 evaluation full and 6th out of 16 in the sub evaluation.

We have learnt that top performance systems have incorporated a couple of semantic concept detectors including audio and visual concepts, which can be more helpful when number of training videos are limited. This is the reason for lower performance of our EK10 system.

On the other hand, our system performed better in EK100 the setting. For example, we got a better performance than the top team in this setting. This indicate that our low level features work well when the number of training videos is abundant. We also observed that our year-to-year improvement on EK10 is 72.5 %, while this number is only 25.9% for the EK100. This observation confirms the contributions of high level and deep learning features in case of event detection with few examplars.

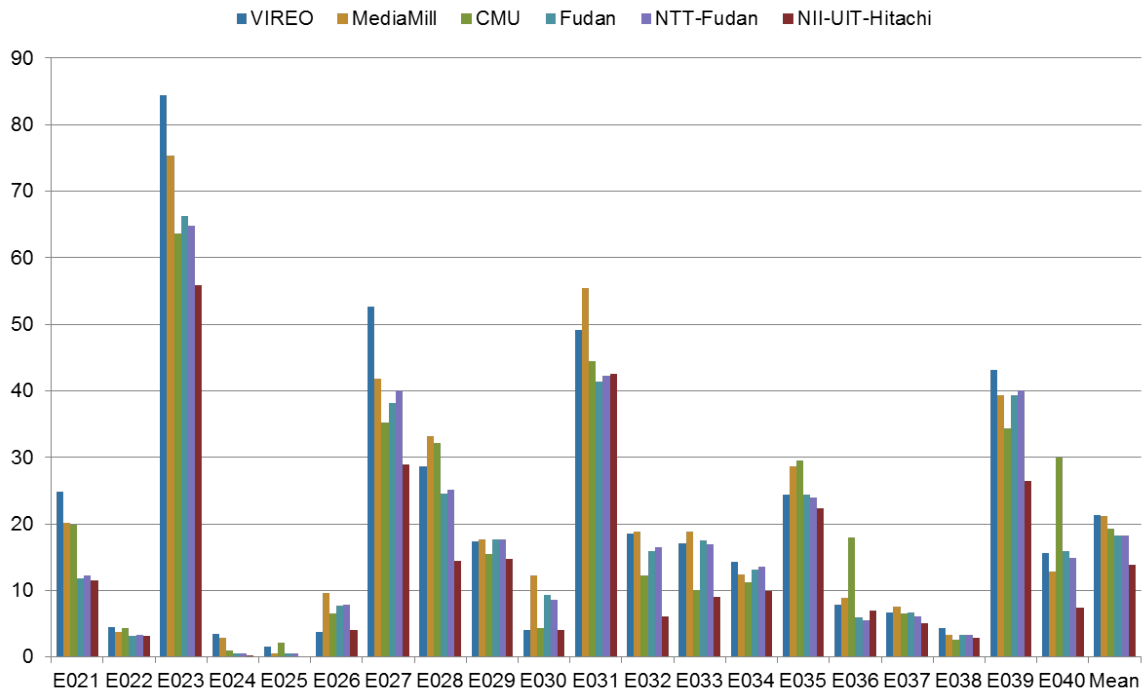


Fig. C.2 Comparison of our performance with top systems in terms of MAP.

Publication List

Journal papers

[1] **S. Phan**, T. D. Ngo, V. Lam, S. Tran, D.-D. Le, D. A. Duong, and S. Satoh. Multimedia event detection using segment-based approach for motion feature. *Journal of Signal Processing Systems*, 74(1):19–31, 2014.

Conference papers

[2] **S. Phan**, D.-D. Le, and S. Satoh. Multimedia event detection using event-driven multiple instance learning. In *ACM Multimedia, 2015*. ACM, 2015.

[3] **S. Phan**, D.-D. Le, and S. Satoh. Sum-max video pooling for complex event recognition. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 1026–1030. IEEE, 2014.

[4] T. D. Ngo, **S. Phan**, D.-D. Le, and S. Satoh. Recommend-me: recommending query regions for image search. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 913–918. ACM, 2014.

[5] T. D. Ngo, V. H. Nguyen, V. Lam, **S. Phan**, D.-D. Le, D. A. Duong, and S. Satoh. Nii-uit: A tool for known item search by sequential pattern filtering. In *MultiMedia Modeling*, pages 419–422. Springer International Publishing, 2014.

[6] V. Lam, **S. Phan**, T. D. Ngo, D.-D. Le, D. A. Duong, and S. Satoh. Violent scene detection using mid-level feature. In *Proceedings of the Fourth Symposium on Information and Communication Technology*, pages 198–205. ACM, 2013.

[7] V. Lam, D.-D. Le, **S. Phan**, S. Satoh, D. A. Duong, and T. D. Ngo. Evaluation of low-level features for detecting violent scenes in videos. In *Soft Computing and Pattern Recognition (SoCPaR), 2013 International Conference of*, pages 213–218. IEEE, 2013.

[8] **S. Phan**, T. D. Ngo, V. Lam, S. Tran, D. Le, D. A. Duong, and S. Satoh. Multimedia event detection using segment-based approach for motion feature. In *Advances in Multimedia Information Processing - PCM 2012 - 13th Pacific-Rim Conference on Multimedia, Singapore, December 4-6, 2012. Proceedings*, pages 33–44, 2012.

Workshop papers

[9] T. Nguyen, **S. Phan**, and T. D. Ngo. Generalized max pooling for action recognition. In *KSE/PR4MCA, 2015*. IEEE, 2015.

[10] D.-D. Le, **S. Phan**, V.-T. Nguyen, C.-Z. Zhu, D. M. Nguyen, T. D. Ngo, S. Kasamwattanarote, P. Sebastien, M.-T. Tran, D. A. Duong, and S. Satoh. National institute of informatics, japan at trecvid 2014. In *TRECVID 2014 Workshop*, 2014.

[11] V. Lam, D. Le, **S. Phan**, S. Satoh, and D. A. Duong. NII-UIT at mediaeval 2014 violent scenes detection affect task. In *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain, October 16-17, 2014.*, 2014.

[12] D.-D. Le, C.-Z. Zhu, **S. Phan**, D. M. Nguyen, V. Q. Lam, D. A. Duong, H. Jegou, and S. Satoh. National institute of informatics, japan at trecvid 2013. In *TRECVID 2013 Workshop*, 2013.

[13] **S. Phan**, D.-D. Le, and S. Satoh. Nii, japan at the first thumos workshop 2013. In *Working Notes of the THUMOS 2013 Workshop, Sydney, Australia, 2013.*, 2013.

[14] V. Lam, D.-D. Le, **S. Phan**, S. Satoh, and D. A. Duong. Nii-uit at mediaeval 2013 violent scenes detection affect task. In *Working Notes Proceedings of the MediaEval 2013 Workshop, Barcelona, Catalunya, Spain, October 18-19, 2013.*, 2013.

[15] V. Lam, D.-D. Le, **S. Phan**, S. Satoh, and D. A. Duong. Nii, japan at mediaeval 2012 violent scenes detection affect task. In *Working Notes Proceedings of the MediaEval 2012 Workshop, Pisa, Italy, October 4-5, 2012.*, 2012.

[16] **S. Phan**, V. Lam, S. Tran, T. D. Ngo, D.-D. Le, and S. Satoh. A codeword visualization tool for dense trajectory feature. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, pages 672–672. IEEE, 2012.

Index