Chapter 5
Evaluating Multimedia Features and Fusion for Example-Based Event Detection

Gregory K. Myers, Cees G. M. Snoek, Ramakant Nevatia, Ramesh Nallapati, Julien van Hout, Stephanie Pancoast, Chen Sun, Amirhossein Habibian, Dennis C. Koelma, Koen E. A. van de Sande and Arnold W. M. Smeulders

G. K. Myers (✉) · J. van Hout · S. Pancoast
SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA
e-mail: gregory.myers@sri.com

J. van Hout
e-mail: julien.vanhout@sri.com

S. Pancoast
e-mail: stephanie.pancoast@sri.com

C. G. M. Snoek · D. C. Koelma · K. E. A. van de Sande · A. W. M. Smeulders
University of Amsterdam (UvA), Science Park 904, P.O. Box 94323, 1098 GH Amsterdam,
The Netherlands
e-mail: cgmsnoek@uva.nl

D. C. Koelma
e-mail: koelma@uva.nl

K. E. A. van de Sande
e-mail: ksande@uva.nl

A. W. M. Smeulders
e-mail: ArnoldSmeulders@uva.nl

R. Nevatia · C. Sun
Institute for Robotics and Intelligent Systems, University of Southern California (USC),
Los Angeles, CA 90089-0273, USA
e-mail: nevatia@usc.edu

C. Sun
e-mail: chensun@usc.edu

R. Nallapati
Thomas J. Watson Research Center, 1101 Kitchawan Road, Yorktown Heights, NY 10598, USA
e-mail: nallapati@us.ibm.com

A. Habibian
University of Amsterdam (UvA)-Euvision, Matrix II, Science Park 400,
1098 XH Amsterdam, The Netherlands
e-mail: a.habibian@uva.nl

B. Ionescu et al. (eds.), *Fusion in Computer Vision*, Advances in Computer
Vision and Pattern Recognition, DOI: 10.1007/978-3-319-05696-8_5,

Abstract Multimedia event detection (MED) is a challenging problem because of the heterogeneous content and variable quality found in large collections of Internet videos. To study the value of multimedia features and fusion for representing and learning events from a set of example video clips, we created SESAME, a system for video SEarch with Speed and Accuracy for Multimedia Events. SESAME includes multiple bag-of-words event classifiers based on single data types: low-level visual, motion, and audio features; high-level semantic visual concepts; and automatic speech recognition (ASR). Event detection performance was evaluated for each event classifier. The performance of low-level visual and motion features was improved by the use of difference coding. The accuracy of the visual concepts was nearly as strong as that of the low-level visual features. Experiments with a number of fusion methods for combining the event detection scores from these classifiers revealed that simple fusion methods, such as arithmetic mean, perform as well as or better than other, more complex fusion methods.

5.1 Introduction

The goal of multimedia event detection (MED) is to detect user-defined events of interest in massive, continuously growing video collections, such as those found on the Internet. This is an extremely challenging problem because the contents of the videos in these collections are completely unconstrained, and the collections include user-generated videos. The quality of such videos varies widely, because they are often made with handheld cameras and may exhibit jerky motions, wildly varying fields of view, and poor lighting. The audio in these videos is recorded in a variety of acoustic environments, often with a single camera-mounted microphone, with no attempt to prevent background sounds from masking speech.

For purposes of this research, an event, as defined in the TREC Video Retrieval Evaluation (TRECVID) MED evaluation task sponsored by the National Institute of Standards and Technology (NIST) [1], has the following characteristics:

- It includes a complex activity occurring at a specific place and time.
- It involves people interacting with other people and/or objects.
- It consists of a number of human actions, processes, and activities that are loosely or tightly organized and have significant temporal and semantic relationships to the overarching activity.
- It is directly observable.

Figure 5.1 shows some sample video imagery from events in the TRECVID MED evaluation task. Events are more complex and may include actions (*hammering*, *pouring liquid*) and activities (*dancing*) occurring in different scenes (*street*, *kitchen*) in indoor and outdoor environments. Some events may be process-oriented, with an expected sequence of stages, actions, or activities (*making a sandwich* or *repairing an appliance*); other events may be a set of ongoing activities with no particular

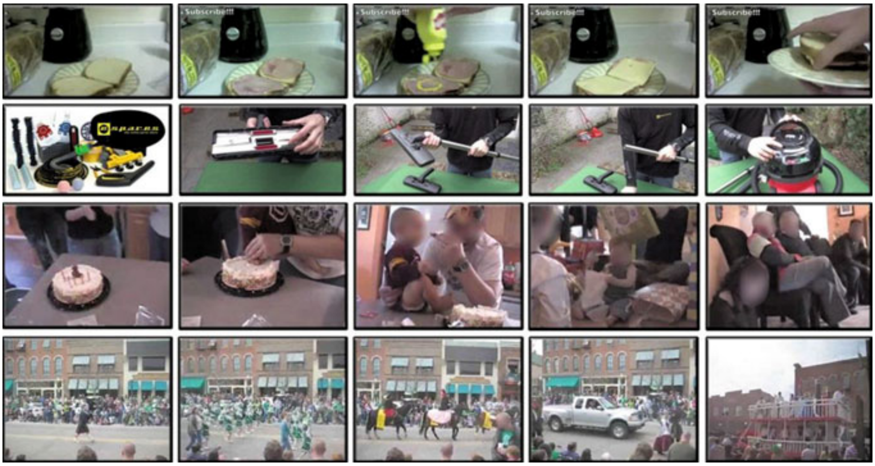

Fig. 5.1 Key frame series from example videos for the events *making a sandwich, repairing an appliance, birthday party,* and *parade* (The imagery was obtained from the Linguistic Data Consortium. Faces have been obscured for privacy)

beginning or end (*birthday party* or *parade*). An event may be observed in only a portion of the video clip, and relevant clips may contain extraneous content.

Multimedia event detection can be considered as a search problem with a query-retrieval paradigm. Currently, videos in online collections, such as YouTube, are retrieved based on text-based search. Text labels are either manually assigned when the video is added to the collection or derived from text already associated with the video, such as text content that occurs near the video in a multimedia blog or web page. Videos are searched and retrieved by matching a text-based user query to videos' text labels, but performance will depend on the quality and availability of such labels.

Highly accurate text-based video retrieval requires the text-based queries to be comprehensive and specific. In the TRECVID MED [1] evaluation, each event is defined by an "**event kit**," which includes a 150–400 word text description consisting of an event name, definition, explication (textual exposition of the terms and concepts), and lists of scenes, objects, people, activities, and sounds that would indicate the presence of the event. Figure 5.2 shows an example for the event w*orking on a woodworking project*. The user might also have to specify how similar events are distinguished from the event of interest (e.g., *not construction* in Fig. 5.2), and may have to estimate the frequency with which various entities occur in the event (e.g., *often indoors*). Subcategories and variations of the event may also have to be considered (e.g., operating a lathe in a factory).

The work described in this chapter focused on evaluating the various data types and fusion methods for MED. The remainder of the paper is organized as follows. A short state-of-the art is presented in Sect. 5.2. Our approach for example-based

Event name: Working on a woodworking project

Definition: One or more people fashion an object out of wood

Explication: Woodworking is a popular hobby that involves crafting an object out of wood. Typical woodworking projects may range from creating large pieces of furniture to small decorative items or toys. The process for making objects out of wood can include cutting wood into smaller pieces with hand or machine tools, carving wood to shape it, sanding wood to smooth it, gluing wood pieces together, drilling holes into wood, and applying decorative finishes to the completed object. Woodworking is distinguished from construction, which typically involves creation of large, permanent structures such as houses, sheds, or buildings, which may or may not be made of wood.

Evidential description:

- **scene:** often indoors in a workshop, garage, artificial lighting, occasionally outdoors
- **objects/people:** woodworking tools (automatic or non-automatic saws, sander, knife), paint, stains, sawhorses, toolbox, safety goggles
- **activities:** cutting and shaping wood, attaching pieces of wood together, smoothing/sanding wood
- **audio:** sounds from power tools, hand tools being used (hammer, saw, etc.); narration of the process

Fig. 5.2 Event kit for *working on a woodworking project*

MED, including methods for content extraction and fusion, is described in Sect. 5.3. Experimental results are described in Sect. 5.4, and Sect. 5.5 contains a summary and discussion.

5.2 Related Work and Motivation for Our Approach

Another approach to detect events is to define the event in terms of a set of example videos, which we call an example-based approach. Example videos are matched to videos in the collection using the same internal representation for each. In this approach, the system automatically learns a model of the event based on a set of positive and negative examples, taking advantage of well-established capabilities in machine learning and computer vision. This chapter considers an example-based approach with both nonsemantic and semantic representations.

Current approaches for MED [2–7] rely heavily on kernel-based classifier methods that use low-level features computed directly from the multimedia data. These classifiers learn a mapping between the computed features and the category of event that occurs in the video. Videos and events are typically represented as "bag-of-words" models composed of histograms of descriptors for each feature type, including visual, motion, and audio features. Although the performance of these models is quite effective, individual low-level features do not correspond directly to terms with semantic meaning, and therefore cannot provide human-understandable evidence of why a video was selected by the MED system as a positive instance of a specific event.

A second representation is in terms of higher-level semantic concepts, which are automatically detected in the video content [8–11]. The detectors are related to objects, like a *flag*; scenes, like a *beach*; people, like *female*; and actions, like *dancing*. The presence of concepts such as these creates an understanding of the content. However, except for a few entities such as faces, most individual concept detectors are not yet reliable [12]. Also, training detectors for each concept require annotated data, which usually involves significant manual effort to generate. In the future, it is expected that more annotated data sets will be available, and weakly supervised learning methods will help improve the efficiency of generating them. Event representations based on high-level concepts have started to appear in the literature [13–16].

For an example-based approach, the central research issue is to find an event representation in terms of the elements of the video that permits the accurate detection of the events. In our approach, an event is modeled as a set of multiple bags-of-words, each based on a single data type. Partitioning the representation by data type permits the descriptors for each data type to be optimized independently. Specific multimodal combinations of features, such as bimodal audiovisual features [3], can be considered a single data type within this architecture. To characterize the video content as comprehensively as possible, the data types we used included a set of heterogeneous low-level features (visual appearance, motion, and audio) and higher-level semantic concepts (visual concepts). We also used automatic speech recognition (ASR) to generate a bag-of-words model in which semantic concepts were expressed directly by words in the recognized speech. The resulting event model combined multiple sources of information from multiple data types and multiple levels of information.

As part of the optimization process for the low-level features, we investigated the use of difference coding techniques in addition to conventional coding methods. Because the information captured by difference coding is somewhat complementary to the information produced by the traditional bag-of-words, we anticipated an improvement in performance. We conducted experiments to compare the performance of difference coding techniques with conventional feature coding techniques.

The remaining challenge is finding the best method for combining the multiple bags-of-words in the event-detection decision process. In the computer vision and multimedia retrieval literature, several fusion methods have been explored [3, 5, 17–19]. For event detection, the most common approach is to apply late fusion methods [3, 5, 17] in which the results for each data type are combined by fusing the decision scores from multiple event classifiers. This is a straightforward way of using the information from all data types in proportion to their relative contribution to event detection on videos with widely diverse content. We evaluated the performance of several late fusion methods.

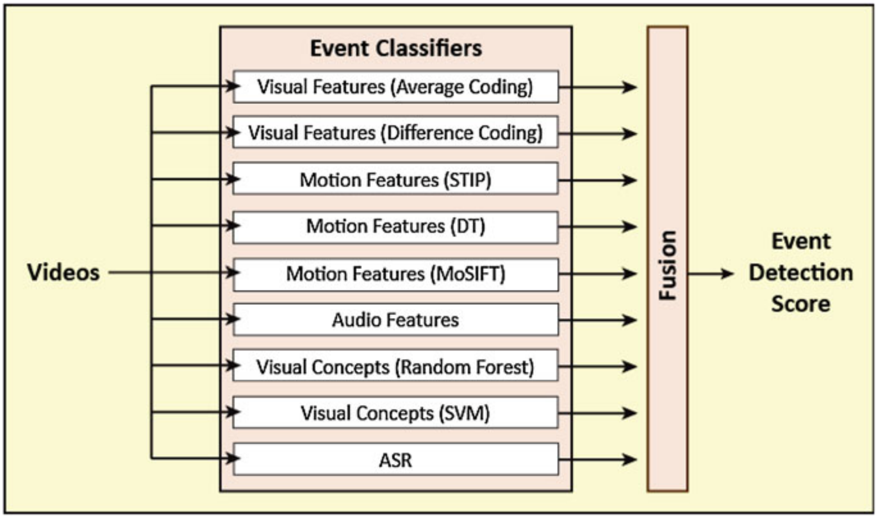

Fig. 5.3 Major components of the SESAME system

All of the experiments for evaluating the performance of the MED capability were performed using the data provided in the TRECVID MED [1] evaluation task. The MED evaluation uses the Heterogeneous Audio Visual Internet Collection (HAVIC) video data collection [20], which is a large corpus of Internet multimedia files collected by the Linguistic Data Consortium.

5.3 Approach for Example-Based MED

The work in this chapter focuses on SESAME, an MED system in which an event is specified as a set of video clip examples. A supervised learning process trains an event model from positive and negative examples, and an event classifier uses the event model to detect the targeted event. An event classifier was built for each data type. The results of all the event classifiers were then combined by fusing their decision scores. An overview of the SESAME system and methods for event classification and fusion are described in the following sections.

5.3.1 SESAME System Overview

The major components of the SESAME system are shown in Fig. 5.3. A total of nine event classifiers generate event detection decision scores: two based on low-level visual features, three based on low-level motion features, one based on low-level audio features, two based on visual concepts, and one based on ASR. (The particular

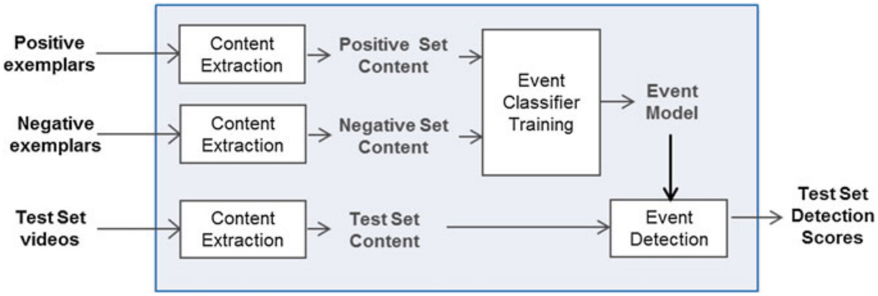

Fig. 5.4 Example-based event classifier for MED

set of classifiers used for each feature type was experimentally determined to be optimal with respect to performance on a reduced dataset.) The outputs of the event classifiers are combined by the fusion process.

Figure 5.4 shows the processing blocks within each event classifier. Each event classifier operates on a single type of data and includes both training and event classification. Content is extracted from positive and negative video examples, and the event classifier is trained, resulting in an event model. The event model produces event detection scores when it is applied to a test set of videos. Figure 5.4 does not show off-line training and testing to optimize the parameter settings for the content extraction processes.

5.3.2 *Content Extraction Methods*

This section describes the feature coding and aggregation methods that were common to the low-level features and the content extraction methods for the different data types: low-level visual features, low-level motion features, low-level audio features, high-level visual features, and ASR.

5.3.2.1 Feature Coding and Aggregation

The coding and aggregation of low-level features share common elements that we describe here. We extracted local features and aggregated them by using three approaches: conventional bag of words (BOW), vector of locally aggregated descriptors (VLAD), and Fisher vectors (FV).

The conventional BOW approach partitions low-level features into clusters to generate a codebook. Given a set of features from a video, a histogram is generated by assigning each feature from the set to one or several nearest code words. Several modifications to this approach are possible. One variation uses soft coding, where instead of assigning each feature to a single code word, distances from the code words are used to weigh the histogram terms for the code words. Another variation describes code words by a Gaussian mixture model (GMM), rather than just by the center of a cluster.

While conventional BOW aggregation has been successfully used for many applications, it does not maintain any information about the distribution of features in the feature space. FV has been introduced in previous work [21] to capture more detailed statistics, and has been applied to image classification and retrieval [22, 23] and capturing variation in time in video [24]. The basic idea is to represent a set of data by a gradient of its log-likelihood to model parameters and to measure the distance between instances with the Fisher kernel. For local features extracted from videos, it becomes natural to model their distribution as GMMs, forming a soft codebook. With GMM, the dimension of FV is linear in the number of mixtures and local feature dimensions.

Finally, VLAD [22] is proposed as a nonprobabilistic version of FV. It uses *k*-means instead of GMM, and accumulates the relative positions of feature points to their single nearest neighbors in the codebook.

Compared with conventional BOW, FV and VLAD have the following benefits:

- FV takes GMM as the underlying generative model.
- Both FV and VLAD are derivatives, so feature points with the same distribution as the general model has no overall impact on the video-level descriptors; as a result, FV and VLAD can suppress noisy and redundant signals.

None of the above aggregation methods consider feature localization in space or in time. We introduced a limited amount of this information by dividing the video into temporal segments (for time localization) and spatial pyramids (for spatial localization). We then compute the features in each segment or block separately and concatenate the resulting features. The spatial pooling and temporal segmentation parameters that yielded the best performance were determined through experimentation.

5.3.2.2 Visual Features

Two event classifiers were developed based on low-level visual features that have proven themselves for general video categorization in TRECVID [25]. They both follow a pipeline consisting of four stages: spatiotemporal sampling of points of interest, visual description of those points, encoding the descriptors into visual words, and supervised learning with kernel machines.

Spatiotemporal Sampling: The visual appearance of an event in video may have a dependency on the spatiotemporal viewpoint under which it is recorded. Salient point methods [26] introduce robustness against viewpoint changes by selecting points, which can be recovered under different perspectives. To determine salient points, Harris-Laplace relies on a Harris corner detector; applying it on multiple scales makes it possible to select the characteristic scale of a local corner using the Laplacian operator. For each corner, the Harris-Laplace detector selects a scale-invariant point if the local image structure under a Laplacian operator has a stable maximum.

Another solution is to use many points by dense sampling. For imagery with many homogenous areas, such as outdoor snow scenes, corners may be rare, so relying on a Harris-Laplace detector can be suboptimal. To counter the shortcomings of Harris-Laplace, we used dense sampling, which samples an image grid in a uniform fashion, using a fixed pixel interval between regions.

In our experiments, we used an interval distance of six pixels and sampled at multiple scales. Appearance variations caused by temporal effects were addressed by analyzing video beyond the key frame level [27]. Taking more frames into account during analysis allowed us to recognize events that were visible during the video, but not necessarily in a single key frame. We sampled one frame every two seconds. Both Harris-Laplace and dense sampling give an equal weight to all keypoints, regardless of their spatial location in the image frame. To overcome this limitation, Lazebnik et al. [28] suggest repeated sampling of fixed subregions of an image, e.g., 1×1, 2×2, 4×4, etc., and then aggregating the different resolutions into a spatial pyramid, which allows for region-specific weighting. Since every region is an image in itself, the spatial pyramid can be combined with both the Harris-Laplace point detector and dense point sampling. We used a spatial pyramid of 1×1 and 1×3 regions, because this was the set of regions that yielded the best performance in our experiments.

Visual Descriptors: In addition to the visual appearance of events in the spatiotemporal viewpoint under which they are recorded, the lighting conditions during recording also play an important role in MED. Properties of color features under classes of illumination and viewing features, such as viewpoint, light intensity, light direction, and light color, can change, specifically for real-world datasets as considered within TRECVID [29]. We followed [25] and used a mixture of SIFT, OpponentSIFT, and C-SIFT descriptors. The SIFT feature proposed by Lowe [30] describes the local contrast of a region using edge-orientation histograms. Because the SIFT feature is normalized, the gradient magnitude changes have no effect on the final feature. OpponentSIFT describes all the channels in the opponent color space using SIFT features. The information in the O3 channel is equal to the intensity information, while the other channels describe the color information in the image. The feature normalization, as effective in SIFT, cancels out any local changes in light intensity. In the opponent color space, the O1 and O2 channels still contain some intensity information. To add invariance to shadow and shading effects, the C-invariant [31] eliminates the remaining intensity information from these channels. The C-SIFT feature uses the C-invariant, which can be seen as the gradient (or derivative) for the normalized opponent color space O1/I and O2/I. The I intensity channel remains unchanged. C-SIFT is known to be scale-invariant with respect to light intensity. We computed the SIFT and C-SIFT descriptors around salient points obtained from the Harris-Laplace detector and dense sampling. We then reduced all descriptors to 80 dimensions with principal component analysis (PCA), a common procedure in the video categorization literature [25].

Word Encoding: To avoid using all low-level visual features from a video, we followed the well-known codebook approach. We first assigned the features to discrete codewords from a predefined codebook. Then, we used the frequency distribution of the codewords as a compact feature vector representing an image frame. Based on [25], we employed codebook construction using *k*-means clustering in combination with average codeword assignment and a maximum of 4,096 codewords. (The number of codewords and the values of other parameters selected for this approach were determined through experimentation.) The traditional hard assignment can be improved by using soft assignment through kernel codebooks [32]. A kernel codebook uses a kernel function to smooth the hard assignment of (image) features to codewords by assigning descriptors to multiple clusters weighted by their distance to the center. We also used difference coding, with VLAD performing *k*-means clustering of the PCA-reduced descriptor space with 1,024 components. The output of the word encoding is a BOW vector using either hard average coding or soft VLAD coding. The BOW vector forms the foundation for event detection.

Kernel Learning: Kernel-based learning methods are typically used to develop robust event detectors from audiovisual features. As described in [25], we relied predominantly on the support vector machine framework for supervised learning of events: specifically, the LIBSVM[1] implementation with probabilistic output. To handle imbalance in the number of positive versus negative training examples, we fixed the weights of the positive and negative classes by estimating the prior probabilities of the classes on training data. We used the histogram intersection kernel and its efficient approximation as suggested by Maji et al. [33]. For difference coded BOWs, we used a linear kernel [21].

Experiments: We evaluated the performance of these two event classifiers on a set of 12,862 drawn from the training and development data from the TRECVID MED [1] evaluation. This SESAME Evaluation dataset consisted of a training set of 8,428 videos and a test set of 4,434 videos sampled from 20 event classes and other classes that did not belong to any of the 20 events. To make good use of the limited number of available positive instances of events, the positives were distributed so that, for each event, there were approximately twice as many positives in the training set as there were in the test set. Separate classifiers were trained for each event based on a one-versus-all paradigm. Table 5.1 shows the performance of the two event classifiers measured by mean average precision (MAP). Color-average coding with a histogram intersection kernel (HIK) Support Vector Machine (SVM) slightly outperformed color-difference soft coding with a linear SVM. For events such as *changing a vehicle tire* and *town hall meeting*, the average HIK was the best event representation. However, for some events, such as *flash mob gathering* and *dog show*, the difference coding was more effective. To study whether the representations complement each other, we also performed a simple average fusion; the results indicate a further increase in event detection performance, improving MAP from 0.342 to 0.358 and giving the best overall performance for the majority of events.

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

Table 5.1 Mean average precision (MAP) **of event classifiers with** low-level visual features and their fusion for 20 TRECVID MED [1] evaluation event classes

Event[a]	Average coding with HIK SVM	Difference coding with linear SVM	Fusion
Birthday_party	**0.275**	0.229	0.261
Changing_a_vehicle_tire	**0.305**	0.269	0.302
Flash_mob_gathering	0.602	**0.644**	0.636
Getting_a_vehicle_unstuck	0.457	**0.496**	0.494
Grooming_an_animal	**0.280**	0.222	0.275
Making_a_sandwich	0.268	0.278	**0.314**
Parade	0.416	0.415	**0.427**
Parkour	**0.464**	0.413	0.450
Repairing_an_appliance	0.486	0.469	**0.498**
Working_on_a_sewing_project	0.378	0.388	**0.400**
Attempting_a_bike_trick	0.398	0.350	**0.408**
Cleaning_an_appliance	**0.138**	0.077	0.135
Dog_show	0.595	**0.651**	0.636
Giving_directions_to_a_location	0.123	0.130	0.134
Marriage_proposal	0.058	**0.093**	0.071
Renovating_a _home	0.229	0.273	**0.285**
Rock_climbing	0.488	0.466	**0.507**
Town_hall_meeting	**0.531**	0.463	0.502
Winning_a_race_without_a_vehicle	0.237	**0.284**	0.263
Working_on_a_metal_crafts_project	0.109	0.134	**0.153**
Mean for all events	**0.342**	**0.337**	**0.358**

[a] Best result per event is denoted in bold

5.3.2.3 Motion Features

Many motion features for activity recognition have been suggested in previous work; [4] provides a nice evaluation of motion features for classifying web videos on the NIST MED 2011 dataset. Based on our analysis of previous work and some small-scale experiments, we decided to use three features: spatio-temporal interest points (STIPs), dense trajectories (DTs) [34], and MoSIFT [35]. STIP features are computed at corner-like locations in the 3-D spatio-temporal volume. Descriptors consist of histograms of gradient and optical flow at these points. This is a very commonly used descriptor; more details may be found in [36]. Dense trajectory features are computed on a dense set of local trajectories (typically computed over 15 frames). Each trajectory is described by its shape and by histograms of intensity gradient, optical flow, and motion boundaries around it. Motion boundary features are somewhat invariant to camera motion. MoSIFT, as its name suggests, uses SIFT feature descriptors; its feature detector is built on motion saliency. STIP and DT were extracted using the default parameters as provided[2]; the MoSIFT features were obtained in the form of coded BOW features.[3]

After the extraction of low-level motion features, we generated a fixed-length video-level descriptor for each video. We experimented with the coding schemes described in Sect. 5.3.2.1 for the STIP and DT features; for MoSIFT, we were able to use BOW features only. We used the training and test sets described above.

We trained separate SVM classifiers for each event and each feature type. Training was based on a one-versus- all paradigm. For conventional BOW features, we used the χ^2 kernel. We used the Gaussian kernel for VLAD and FV. To select classifier-independent parameters (such as the codebook size), we conducted fivefold cross validation of 2,062 videos from 15 event classes. We conducted fivefold cross validation on the training set to select classifier-dependent parameters. For BOW features, we used 1,000 codewords; for FV and VLAD, we used 64 cluster centers. More details of the procedure are found in [37].

We compared the performance of conventional BOW, FV, and VLAD for STIP features; BOW and FV for DT features; and BOW for MoSIFT, using the SESAME Evaluation dataset. Table 5.2 shows the results.

We can see that FV gave the best MAP for both STIP and DT. VLAD also improved MAP for STIP, but was not as effective as the FV features. We were not able to perform VLAD and FV experiments for MoSIFT features, but would expect to have seen similar improvements there.

5.3.2.4 Audio Features

The audio is modeled as a *bag of audio words* (BOAW). The BOAW has recently been used for audio document retrieval [38] and copy detection [39], as well as MED tasks [40]. Our recent work [41] describes the basic BOAW approach. We extracted the audio data from the video files and converted them to a 16 kHz sampling rate. We extracted Mel frequency cepstral coefficients (MFCCs) for every 10 ms interval using a hamming window with 50 % overlap. The features consist of 13 values (12 coefficients and the log-energy), along with their delta and delta-delta values. We used a randomized sample of the videos from the TRECVID 2011 MED evaluation development set to generate the codebook. We performed *k*-means clustering on the MFCC features to generate 1,000 clusters. The centroid for each cluster is taken as a code word. The soft quantization process used the codebook to map the MFCCs to code words. We trained an SVM classifier with a histogram intersection kernel on the soft quantization histogram vectors of the video examples, and used the classifier to detect the events. Evaluation with the SESAME Evaluation dataset showed that the audio features achieved a MAP of 0.112.

[2] We obtained the STIP code from http://www.di.ens.fr/~laptev/download/stip-1.1-winlinux.zip, and DT code from http://lear.inrialpes.fr/people/wang/dense_trajectories.

[3] MoSIFT features were provided by Dr. Alex Hauptmann of Carnegie-Mellon University.

Table 5.2 Mean average precision of event classifiers with motion features for 20 TRECVID MED [1] evaluation event classes

Event[a]	BOW + MoSIFT	BOW + STIP	VLAD + STIP	FV + STIP	BOW + DT	FV + DT
Birthday_party	0.191	0.217	0.217	0.189	0.225	**0.293**
Changing_a_vehicle_tire	0.126	0.064	0.165	0.136	0.190	**0.217**
Flash_mob_gathering	0.463	0.535	**0.579**	0.569	0.564	0.567
Getting_a_vehicle_unstuck	0.337	0.284	0.316	0.365	0.403	**0.439**
Grooming_an_animal	**0.290**	0.093	0.116	0.147	0.216	0.247
Making_a_sandwich	0.164	0.154	0.193	0.225	0.198	**0.234**
Parade	0.326	0.260	0.364	**0.457**	0.446	0.419
Parkour	0.295	0.366	0.404	0.369	0.413	**0.459**
Repairing_an_appliance	0.368	0.357	0.370	0.385	0.417	**0.443**
Working_on_a_sewing_project	0.270	0.292	0.346	0.386	0.352	**0.433**
Attempting_a_bike_trick	**0.640**	0.104	0.234	0.235	0.245	0.438
Cleaning_an_appliance	**0.090**	0.058	0.088	0.074	0.066	0.089
Dog_show	0.488	0.361	0.489	0.557	0.600	**0.632**
Giving_directions_to_a_location	0.085	**0.194**	0.148	0.191	0.069	0.052
Marriage_proposal	0.027	0.040	0.107	**0.173**	0.059	0.118
Renovating_a_home	0.157	0.182	0.201	0.255	0.277	**0.361**
Rock_climbing	0.465	0.156	0.326	0.352	**0.470**	0.425
Town_hall_meeting	**0.519**	0.285	0.286	0.462	0.317	0.370
Winning_a_race_without_a_vehicle	**0.273**	0.187	0.174	0.260	0.179	0.216
Working_on_a_metal_crafts_project	0.116	**0.148**	0.064	0.032	0.072	0.128
MAP	**0.285**	**0.217**	**0.259**	**0.291**	**0.289**	**0.329**

[a] Best result per event is denoted in bold

5.3.2.5 Visual Concepts

Two event classifiers were based on concept detectors. We followed the pipeline proposed in [42]. We decoded the videos by uniformly extracting one frame every 2 sec. We then applied all available concept detectors to the extracted frames. After we concatenated the detector outputs, each frame was represented by a concept vector. Finally, we aggregated the frame representations into a video-level representation by averaging and normalization. On top of this concept representation per video, we used either a HIK SVM or a random forest as an event classifier.

To create the concept representation, we needed a comprehensive pool of concept detectors. We built this pool of detectors using the human-annotated training data from two publicly available resources: the TRECVID 2012 Semantic Indexing task [43] and the ImageNet Large-Scale Visual Recognition Challenge 2011 [44]. The former has annotations for 346 semantic concepts on 400,000 keyframes from web videos. The latter has annotations for 1,000 semantic concepts on 1300,000 photos. The categories are quite diverse and include concepts from various types; i.e., objects like *helicopter* and *harmonica*, scenes like *kitchen* and *hospital*, and actions like *greeting* and *swimming*. Leveraging the annotated data available in these datasets, we trained 1,346 concept detectors in total.

Table 5.3 Mean average precision of event classifiers with visual concept features for 20 TRECVID MED [1] evaluation event classes

Event[a]	RF	SVM
Birthday_party	**0.339**	0.324
Changing_a_vehicle_tire	**0.251**	0.241
Flash_mob_gathering	0.542	0.542
Getting_a_vehicle_unstuck	**0.454**	0.426
Grooming_an_animal	**0.254**	0.231
Making_a_sandwich	**0.283**	0.257
Parade	**0.373**	0.306
Parkour	**0.550**	0.479
Repairing_an_appliance	**0.422**	0.404
Working_on_a_sewing_project	0.390	**0.394**
Attempting_a_bike_trick	**0.475**	0.472
Cleaning_an_appliance	0.097	**0.149**
Dog_show	**0.595**	0.529
Giving_directions_to_a_location	0.058	**0.097**
Marriage_proposal	0.077	0.066
Renovating_a_home	0.295	**0.325**
Rock_climbing	**0.412**	0.401
Town_hall_meeting	0.411	**0.417**
Winning_a_race_without_a_vehicle	**0.198**	0.167
Working_on_a_metal_crafts_project	0.099	**0.162**
Mean for all events	**0.341**	**0.330**

[a] Best result per event is denoted in bold

We followed the state-of-the-art for our implementation of the concept detectors. We used densely sampled SIFT, OpponentSIFT, and C-SIFT descriptors, as we had for our event detector using visual features, but this time, we used difference coding with Fisher vectors [21]. We used a visual vocabulary of 256 words. We again used the full image and three horizontal bars as a spatial pyramid. The feature vectors representing the training images formed the input for a linear SVM.

Experiments with the SESAME Evaluation dataset, summarized in Table 5.3, show that the random forest classifier is more successful than the nonlinear HIK SVM for event detection using visual concepts, although the two approaches are quite close on average. Note that the event detection results using visual concepts are close to our low-level representation using visual or motion features.

5.3.2.6 Automatic Speech Recognition

Spoken language content is often present in user-generated videos and can potentially contribute useful information for detecting events. The recognized speech has direct semantic information that typically complements the information contributed by low-level visual features. We used DECIPHER, SRI's ASR software, to recognize spoken English. We used acoustic and language models obtained from an ASR system

[45] trained on speech data recorded in meetings with a far-field microphone. Initial tests on the audio in user-generated videos revealed that the segmentation process, which distinguishes speech from other audio, often misclassified music as speech. Therefore, before running the speech recognizer on these videos, we constructed a new segmenter, which is described below.

The existing segmenter was GMM-based and had two classes (speech and non-speech). For this effort, we leveraged the availability of annotated TRECVID video data and built a segmenter better tuned to audio conditions in user-generated videos. We built a segmenter with four classes: speech, music, noise, and pause. We measured the effectiveness of the new segmentation by the word-error rates (WERs) obtained by feeding the speech-segmented audio to our ASR system. We found that the new segmentation helped reduce the WER from 105 to 83 %. This confirmed that the new segmentation models were a better match to the TRECVID data than models trained on meeting data. For reference, when all the speech segments were processed by the ASR, the WER obtained by our system was 78 %. (This oracle segmentation provided the lowest WER that could be achieved by improving the segmentation.)

To create features for the event classifiers, we used ASR recognition lattices to compute the expected word counts for each word and each video. This approach provided significantly better results compared to using the 1-best ASR output, because it compensated for ASR errors by including words with lower posteriors that weren't necessarily present in the 1-best. We computed the logarithm of the counts for each word, appended them to form a feature vector of dimension 34,457, and used a linear SVM for the event classifiers. More details may be found in [46]. Evaluation with the SESAME Evaluation dataset showed that the ASR event classifiers achieved a MAP of 0.114.

5.3.3 Fusion

We implemented a number of late fusion methods, all of which involved a weighted average of the detection scores from the individual event classifiers. The methods for determining the weights considered several factors:

- **Event dependence and learned weights:** Because the set of most reliable data types for different events might vary, we considered the importance of learning the fusion weights for each event using a training set. However, when there is limited data available for training, aggregating the data for all events and computing a fixed set of weights for all events may yield more reliable results. Another strategy is to set the weights without training with any data at all. For example, in the method of fusing with the arithmetic mean of the scores, all of the weights are equal.
- **Score dependence:** For weights learned via cross-validation on a training set, a single set of fixed weights might be learned for the entire range of detection scores. Alternatively, the multidimensional space of detection scores might be partitioned into a set of regions, with a set of weights assigned to each region. In general, more data is needed for score-dependent weights to avoid overfitting.

- **Adjustment for missing scores:** When the scores for some types of data (particularly for ASR and MFCC) are missing, a default value, such as an average for the missing score, might be used, but this could provide a misleading indication of contribution and therefore degrade performance. Another way in which the fusion methods described below dealt with missing scores was renormalizing the weights of the nonmissing scores. Another alternative was learning multiple sets of weights, each set for a particular combination of nonmissing scores.

We evaluated the fusion models described below. All of the models operated on detection scores that were normalized using a Gaussian function (i.e., computing the z-score by removing the mean and scaling by the standard deviation).

Arithmetic mean (AM): In this method, we compute the AM of the scores of the observed data types for a given clip. Missing data types for a given clip are ignored, and the averaging is performed over the scores of observed data types.

Geometric mean (GM): In this method, we compute the uniform GM of the scores of the observed data types for a given clip. As we do for AM, we ignore missing data types and compute the geometric mean of the scores from observed data types.

Mean average precision-weighted fusion (MAP): This fusion method weighs scores from the observed data types for a clip by their normalized average precision scores, as computed on the training fold. Again, the normalization is performed only over the observed data types for a given clip.

Weighted mean root (WMR): This fusion method is a variant of the MAP-weighted method. In this method, we compute the fusion score as we do for MAP-weighted fusion, except the final fused score $\boldsymbol{x}'$ is determined by performing a power normalization of the MAP-based fused score $\boldsymbol{x}$:

$$\boldsymbol{x}' = \boldsymbol{x}^{\frac{1}{\alpha}} \tag{5.1}$$

where α is the number of nonmissing data types for that video. In other words, the higher the number of data types from which the fusion score is computed, the more trustworthy the output.

Conditional mixture model: This model combines the detection scores from various data types using mixture weights that were trained by the expectation maximization (EM) algorithm on the labeled training folds. For clips that are missing scores from one or more data types, we provide the expected score for that data type based on the training data.

Sparse mixture model (SMM): This extension of the conditional mixture model addresses the problem of missing scores for a clip by computing a mixture for only the observed data types [47]. This is done by renormalizing the mixture weights over the observed data types for each clip. The training was done with the EM algorithm, but the maximization step no longer had a closed-form solution, so we used gradient-descent techniques to learn the optimal weights.

Table 5.4 Fusion methods and their characteristics

Fusion method	Event-independent?	Learned on a training set?	Score-dependent?	Adjustment for missing scores?
Arithmetic mean	Yes	No	No	Yes
Geometric mean	Yes	No	No	Yes
MAP-weighted	No	Yes	No	Yes
Weighted mean root	No	Yes	No	Yes
Conditional mixture model	No	Yes	No	No
Sparse mixture model	No	Yes	No	Yes
SVMLight	No	Yes	Yes	No
Distance from threshold	No	Yes	Yes	No
Bin accuracy weighting	No	Yes	Yes	No

SVMLight: This fusion model consists of training an SVM using the scores from various data types as the features for each clip. Missing data types for a given clip are assigned zero scores. We used the SVMLight[4] implementation with linear kernels.

Distance from threshold: This is a weighted averaging method [3] that dynamically adjusts the weights of each data type for each video clip based on how far the score is from its decision threshold. If the detection score is near the threshold, the correct decision is presumed to be somewhat uncertain, and a lower weight is assigned. A detection score that is much greater or much lower than the threshold indicates that more confidence should be placed in the decision, and a higher weight is assigned.

Bin accuracy weighting: This method tries to address the problem of uneven distribution of detection scores in the training set. For each data type, the range of scores in the training fold is divided into bins with approximately equal counts per bin. During training, the accuracy of each bin is measured by computing the proportion of correctly classified videos whose scores fall within the bin. During testing, for each data type, the specific bin that the scores fall into is determined, and the corresponding bin accuracy scores for each data type are used as fusion weights.

Table 5.4 summarizes the fusion methods and their characteristics.

5.4 Experimental Results

We evaluated the performance of our SESAME system using the data provided in the TRECVID MED [1] evaluation task. Although the MED event kit contained both a text description and video examples for each event, the SESAME system implemented the example-based approach in which only the video examples were used for event detection training.

[4] http://svmlight.joachims.org/

5.4.1 Evaluation by Data Type

Table 5.5 lists results on the SESAME Evaluation dataset, which consisted of a training set of 8,428 videos and a test set of 4,434 videos, sampled from 20 event classes and other classes that did not belong to any of the 20 events. In terms of the performance of the various data types, the visual features were the strongest performers across all events. The accuracy of the visual concepts was nearly as strong as that of the low-level visual features. The motion features also showed strong performance. Although the performance of low-level audio features and ASR was significantly less, ASR had the highest performance for events containing a relatively large amount of speech content, including a number of instructional videos. The best scores for each event are distributed among all of the data types, indicating that fusion of these data should yield improved performance. Indeed, the AM fusion of the individual event classifiers, which is listed in the last column of Table 5.5, shows a significant boost in performance: a 33 % improvement over the best single data type.

5.4.2 Evaluation of Fusion Methods

We tested the late fusion methods described in Sect. 5.3.3 using the SESAME Evaluation dataset. For all our fusion experiments, we trained each event classifier on the training set, and executed the classifier on the test set to produce detection scores for each event. To produce legitimate fusion scores over the test set, we used tenfold cross validation, with random fold selection, to generate the detections, and then obtained a micro-averaged average precision over the resulting detections. The micro-averaged MAP was computed by averaging the average precision for each event. To gauge the stability of the fusion methods, we repeated this process 30 times and computed the macro average and standard deviation of the micro-averaged MAPs. Because the Arithmetic Mean and Geometric Mean methods are untrained, their micro-averaged MAPs will be the same regardless of fold selection; thus, the standard deviations for their micro-averaged MAPs are zero.

Table 5.6 shows the MED performance of various fusion methods. The comparison indicates that the simplest fusion methods, such as AM and GM, performed as well as or better than other, more complex fusion methods. Also note that most of the top-performing fusion methods (AM, GM, MAP, WMR, and SMM) adjusted their weights to accommodate missing scores.

5.4.3 Evaluation of MED Performance in TRECVID

As the SESAME team, we participated in the 2012 TRECVID MED [1] evaluation and submitted the detection results for a system configured nearly the same as that

Table 5.5 Experiment results in terms of mean average precision for individual event classifiers

Event[a]	Low-level visual features		Visual concept features		Motion features			Audio		Fusion
	SIFT-AVG	SIFT-DC	RF	SVM	STIP	DT	MOSIFT	MFCC	ASR	AM
Birthday_party	0.275	0.229	**0.339**	0.324	0.189	0.293	0.191	0.146	0.062	0.372
Changing_a_vehicle_tire	**0.305**	0.270	0.251	0.241	0.136	0.217	0.126	0.024	0.209	0.343
Flash_mob_gathering	0.603	**0.644**	0.542	0.542	0.569	0.567	0.463	0.139	0.017	0.644
Getting_a_vehicle_unstuck	0.457	**0.496**	0.454	0.426	0.365	0.439	0.337	0.040	0.011	0.586
Grooming_an_animal	0.280	0.222	0.254	0.231	0.147	0.247	**0.290**	0.038	0.024	0.352
Making_a_sandwich	0.267	0.278	0.283	0.257	0.225	0.234	0.164	0.038	**0.378**	0.392
Parade	0.416	0.414	0.373	0.306	**0.457**	0.419	0.326	0.119	0.013	0.578
Parkour	0.464	0.414	**0.550**	0.479	0.369	0.459	0.295	0.029	0.009	0.564
Repairing_an_appliance	0.486	0.469	0.422	0.404	0.385	0.443	0.368	0.449	**0.517**	0.591
Working_on_a_sewing_project	0.378	0.388	0.390	**0.394**	0.386	0.433	0.270	0.192	0.276	0.551
Attempting_a_bike_trick	0.398	0.350	0.475	0.472	0.235	0.438	**0.640**	0.019	0.003	0.703
Cleaning_an_appliance	0.138	0.077	0.097	**0.149**	0.074	0.089	0.090	0.050	0.144	0.174
Dog_show	0.591	**0.650**	0.595	0.529	0.557	0.632	0.488	0.183	0.002	0.672
Giving_directions_to_a_location	0.123	0.130	0.058	0.097	**0.191**	0.052	0.085	0.075	0.066	0.193
Marriage_proposal	0.057	0.093	0.077	0.066	**0.173**	0.118	0.027	0.044	0.010	0.179
Renovating_a_home	0.229	0.273	0.295	0.325	0.255	**0.361**	0.157	0.099	0.145	0.461
Rock_climbing	**0.488**	0.466	0.412	0.401	0.352	0.425	0.465	0.020	0.005	0.615
Town_hall_meeting	**0.531**	0.463	0.411	0.417	0.462	0.370	0.519	0.433	0.341	0.649
Winning_a_race_without_a_vehicle	0.237	**0.284**	0.198	0.167	0.260	0.216	0.273	0.074	0.005	0.295
Working_on_a_metal_crafts_project	0.109	0.133	0.099	**0.162**	0.032	0.128	0.116	0.024	0.044	0.209
Mean for all events	**0.342**	**0.337**	**0.341**	**0.330**	**0.291**	**0.329**	**0.285**	**0.112**	**0.114**	**0.456**

[a] The data type with the highest MAP score for each event is in bold. The AM fusion of the individual event classifiers is listed in the last column

Table 5.6 MED performance of fusion methods with all event classifiers

Fusion method	MacroMAP	Standard deviation
Arithmetic mean	0.456	0.0000
Geometric mean	0.456	0.0000
MAP-weighted	0.437	0.0006
Weighted Mean Root	0.451	0.0005
Conditional mixture model	0.403	0.0054
Sparse mixture model	0.443	0.0007
SVMLight	0.451	0.0036
Distance from threshold	0.407	0.0005
Bin accuracy weighting	0.401	0.0031

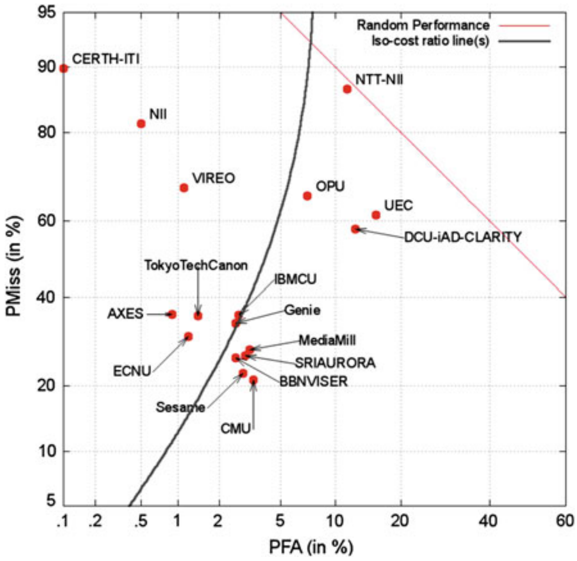

Fig. 5.5 Performance of the primary runs of 17 MED systems in the 2012 TRECVID MED [1] evaluation

described in this chapter.[5] The event classifiers were trained with all the positives from the event kit and negatives from the TRECVID MED training and development material. The test set consisted of the 99,000 videos used in the formal evaluation.

Figure 5.5 shows the performance of the primary runs of 17 MED systems in this evaluation in terms of miss and false alarm rates [48]. The performance of the SESAME run was one of the best among the evaluation participants.

[5] It included a poorer-performing ASR capability instead of the one described in Sect. 5.3.2.6, and a video OCR capability that contributed minimally to overall performance.

5.5 Summary and Discussion

SESAME, a MED capability that learns event models from a set of example video clips, includes a number of BOW event classifiers based on single data types: low-level visual, motion, and audio features; high-level semantic visual concepts; and ASR. Partitioning the representation by data type permits the descriptors for each data type to be optimized independently. We evaluated the detection performance for each event classifier and experimented with a number of fusion methods for combining the event detection scores from these classifiers. Our experiments using multiple data types and late fusion of their scores demonstrated strongly reliable MED performance.

Major conclusions from this effort include:

- The relative contribution of visual, motion, and audio features varies according to the specific event. This is due to differences in the relative distinctiveness and consistency of certain features for each event category. Across all events, score-level fusion resulted in a 33 % improvement over the best single data type, indicating that different types of features contribute to the representation of heterogeneous video data.
- The use of difference coding in low-level visual and motion features significantly improved performance. We surmise that difference coding works better than the traditional bag-of-words because it measures differences from the general model, which is likely to be dominated by the background features. We expect additional gains in performance if difference coding were applied to low-level audio features.
- The set of 1346 high-level visual features was nearly as effective as the set of low-level visual features. It appears that, in comparison to the 5000 or so concepts predicted to be needed for sufficient performance in event detection [49], this number of high-level features begins to span the space of concepts reasonably well. Therefore, analogous sets of motion and audio concepts should further improve overall performance.
- Although the performance of ASR was lower than that of the visual and motion features, its performance was highly event-dependent, and it performed reasonably well for events containing a relatively large amount of speech content, such as instructional videos.
- The simplest fusion methods for computing event detection scores were very effective compared to more complex fusion methods. One possible explanation for this is that the reliability of the scores is roughly equal across all data types. Another possible reason is that the limited number of positive training examples (an average of about 70 per event) is not enough to achieve the full benefit of the more complex fusion models.

While our relatively straightforward BOW approach was quite effective, we view it as a baseline capability that could be improved in several ways:

- Since the current approach aggregates low-level visual and motion features within fixed spatial partitions, the usage of local information is limited. Features of an object divided by our predefined partition, for example, will not be aggregated as a whole. We expect that the use of dynamic spatial pooling, which is better aligned to the structure and content of the video imagery, will improve performance. Segmenting the image into meaningful homogeneous regions would be even better, as it allows more salient characteristics to be extracted, and would eventually lead to better classification.
- The current approach ignores the temporal information within each video clip; all of the visual, motion, and audio features are aggregated. However, events consist of multiple components that appear at different times, so using time-based information for event modeling and detection should improve performance. Also, aggregating low-level features according to the temporal structure of the video may yield feature sets that better represent the video contents.
- All of the classifiers in our approach operate on a histogram of features and do not leverage any relationships between the features. Features occurring in video data are not generally independent. In particular, the combination of particular high-level semantic concepts could become strong discriminatory evidence, since their co-occurrence might be associated with a subset of relevant video content. For example, although the concepts *balloons* and *singing* occur in many contexts, the occurrence of both might be more common to *birthday party* than to other video content. Exploiting the spatiotemporal dependencies among the features would better characterize the video contents and offer a richer set of data with which to build event models.

Acknowledgments This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via the Department of Interior National Business Center, contract number D11PC0067. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

1. Smeaton AF, Over P, Kraaij W (2006) Evaluation campaigns and TRECVID. In: Proceedings of the 8th ACM international workshop on multimedia information retrieval (MIR '06), Santa Barbara, 26–27 October 2006, ACM Press, New York, pp 321–330
2. Jiang Y-G, Bhattacharya S, Chang S-F, Shah M (2012) High-level event recognition in unconstrained videos. Int J Multimedia Inf Retrieval 2:1–29
3. Natarajan P, Wu S, Vitaladevuni S, Zhuang X, Tsakalidis S, Paurk U, Prasad R (2012) Multimodal feature fusion for robust event detection in web videos. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR), pp 1298–1305

4. Sawhney H, Cheng H, Divakaran A, Javed O, Liu J, Yu Q, Ali S, Tamrakar A (2012) Evaluation of low-level features and their combinations for complex event detection in open source videos. In: CVPR, pp 2496–2499
5. Jiang Y (2013) Super: towards real-time event recognition in internet videos. In: ACM international conference on multimedia retrieval (ICMR), Article No. 33
6. Ballan L, Bertini M, Del Bimbo A, Seidenari L, Serra G (2011) Event detection and recognition for semantic annotation of video. Multimedia Tools Appl 51(1):279–302
7. Xu D, Chang S-F (2008) Video event recognition using kernel methods with multilevel temporal alignment. IEEE Trans Pattern Anal Mach Intell 30(11):1985–1997
8. Snoek CGM, Worring M (2009) Concept-based video retrieval. Found Trends Inf Retr 2(4):214–322
9. Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE TPAMI 32(9):1627–1645
10. Li L, Su H, Xing E, Fei-Fei L (2010) Object bank: a high-level image representation for scene classification and semantic feature sparsification. In: Advances in Neural Information Processing Systems, p. 24
11. Sadanand S, Corso JJ (2012) Action bank: a high-level representation of activity in video. In: CVPR
12. Snoek CGM, Smeulders AWM (2010) Visual-concept search solved? IEEE Comput 43(6):76–78
13. Merler M, Huang B, Xie L, Hua G, Natsev A (2012) Semantic model vectors for complex video event recognition. IEEE Trans Multimedia (TMM) 14(1):88–101
14. Althoff T, Song H, Darrell T (2012) Detection bank: an object detection based video representation for multimedia event recognition. In: ACM Multimedia (MM)
15. Tsampoulatidis I, Gkalelis N, Dimou A, Mezaris V, Kompatsiaris I (2011) High-level event detection in video exploiting discriminant concepts. In: Proceedings of the 1st ACM international conference on multimedia retrieval, p 8590
16. Habibian A, van de Sande KEA, Snoek CGM (2013) Recommendations for video event recognition using concept vocabularies. In: Proceedings of the ACM international conference on multimedia retrieval, Dallas, pp 89–96
17. Perera AGA, Oh S, Leotta M, Kim I, Byun B, Lee CH, McCloskey S, Liu J, Miller B, Huang ZF, Vahdat A, Yang W, Mori G, Tang K, Koller D, Fei-Fei L, Li K, Chen G, Corso J, Fu Y, Srihari R (2011) GENIE TRECVID 2011 multimedia event detection: late-fusion approaches to combine multiple audio-visual features. In: NIST TRECVID workshop
18. Pinquier J, Karaman S, Letoupin L, Guyot P, Mégret R, Benois-Pineau J, Gaëstel Y, Dartigues J-F (2012) Strategies for multiple feature fusion with Hierarchical HMM: application to activity recognition from wearable audiovisual sensors. In: International conference on pattern recognition, pp 3192–3195
19. Csurka G, Clinchant S (2012) An empirical study of fusion operators for multimodal image retrieval. In: Content-based multimedia indexing, pp 1–6
20. Strassel S, Morris A, Fiscus J, Caruso C, Lee H, Over P, Fiumara J, Shaw B, Antonishek B, Michel M (2012) Creating HAVIC: heterogeneous audio visual internet collection. In: Calzolari N, Choukri K, Declerck T, Uğur Doğan M, Maegaard B, Mariani J, Odijk J, Piperidis S (eds) Proceedings of the eighth international conference on language resources and evaluation, Istanbul
21. Jaakkola T, Haussler D (1999) Exploiting generative models in discriminative classifiers. In: Proceedings of the 1998 conference on advances in neural information processing systems II, pp 489–493
22. Jégou H, Perronnin F, Douze M, Sanchez J, Pérez P, Schmid C (2012) Aggregating local image descriptors into compact codes. IEEE TPAMI 34(9):1704–1716
23. Perronnin F, Dance C (2007) Fisher kernels on visual vocabularies for image categorization. In: CVPR
24. Mironica I, Uijlings J, Rostamzadeh N, Ionescu B, Sebe N (2013) Time matters!: capturing variation in time in video using fisher kernels. In: ACM multimedia, pp 701–704

25. Snoek CGM, van de Sande KEA, Habibian A, Kordumova S, Li Z, Mazloom M, Pintea SL, Tao R, Koelma DC, Smeulders AWM (2012) The MediaMill TRECVID 2012 semantic video search engine. In: Proceeding of TRECVID Workshop, Gaithersburg
26. Tuytelaars T, Mikolajczyk K (2008) Local invariant feature detectors: a survey. Found Trends Comp Graphics Vision 3(3):177–280
27. Snoek CGM, Worring M, Geusebroek J-M, Koelma DC, Seinstra FJ (2005) On the surplus value of semantic video analysis beyond the key frame. In: Proceedings of the IEEE international conference on multimedia and expo
28. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: CVPR, New York, vol 2. pp 2169–2178
29. van de Sande KEA, Gevers T, Snoek CGM (2010) Evaluating color descriptors for object and scene recognition. IEEE TPAMI 32(9):1582–1596
30. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60:91–110
31. Geusebroek J-M, Boomgaard R, Smeulders AWM, Geerts H (2001) Color invariance. IEEE TPAMI 23(12):13381350
32. van Gemert JC, Snoek CGM, Veenman CJ, Smeulders AWM, Geusebroek J-M (2010) Comparing compact codebooks for visual categorization. Comput Vis Image Und 114(4):450–462
33. Maji S, Berg AC, Malik J (2008) Classification using intersection kernel support vector machines is efficient. In: Proceedings of the IEEE computer society conference on CVPR, Anchorage, pp 619–626
34. Wang H, Kläser A, Schmid C, Cheng-Lin L (2011) Action recognition by dense trajectories. In: CVPR, pp 3169–3176
35. Chen M-Y, Hauptmann A (2009) MoSIFT: recognizing human actions in surveillance videos. In: CMU-CS-09-161, Carnegie Mellon University
36. Laptev I (2005) On space-time interest points. Int J of Comput Vision 64(2/3):107–123
37. Sun C, Nevatia R (2013) Large scale web video classification by use of Fisher vectors. In: Workshop on applications of computer vision, Clearwater
38. Chechik G, Ie E, Rehn M, Bengio S, Lyon D (2008) Large-scale content-based audio retrieval from text queries. In: Proceedings of 1^{st} ACM international conference on multimedia information retrieval (MIR '08), New York, pp 105–112
39. Uchida Y, Sakazawa S, Argawal M, Akbacak M (2010) KDDI labs and SRI international at TRECVID 2010: content-based copy detection. In: NIST TRECVID 2010 evaluation, workshop
40. Jiang Y, Zeng X, Ye G, Ellis D, Shah M, Chang S (2010) Columbia-UCF TRECVID 2010 multimedia event detection: combining multiple modalities, contextual concepts, and temporal matching. In: NIST TRECVID Workshop
41. Pancoast S, Akbacak M (2012) Bag-of-audio-words approach for multimedia event detection. In: Proceedings of interspeech
42. Merler M, Huang B, Xie L, Hua G, Natsev A (2012) Semantic model vectors for complex video event recognition. IEEE Trans Multimedia 14(1):88101
43. Over P, Awad G, Michel M, Fiscus J, Sanders G, Shaw B, Kraaij W, Smeaton AF, Quéenot G (2012) TRECVID 2012: an overview of the goals, tasks, data, evaluation mechanisms, and metrics. In: Proceedings of TRECVID. http://www-nlpir.nist.gov/projects/tvpubs/tv12.papers/tv12overview.pdf
44. Berg A, Deng J, Satheesh S, Su H, Li F-F (2011) Imagenet large scale visual recognition challenge. http://www.image-net.org/challenges/LSVRC/2011/
45. Janin A, Stolcke A, Anguera X, Boakye K, Çetin Ö, Frankel J, Zheng J (2006) The ICSI-SRI spring 2006 meeting recognition system. In: MLMI'06 proceedings of the third international conference on machine learning for multimodal, interaction, pp 444–456
46. van Hout J, Akbacak M, Castaneda D, Yeh E, Sanchez M (2013) Extracting audio and spoken concepts for multimedia event detection. In: International conference on acoustics, speech, and signal processing (ICASSP)

47. Nallapati R, Yeh E, Myers G (2012) Sparse mixture model: late fusion with missing scores for multimedia event detection. Algorithms and systems VII, SPIE multimedia content access
48. Fiscus J, Michel M (2012) TRECVID 2012 multimedia event detection task. In: NIST TRECVID 2012 evaluation, workshop
49. Hauptmann A, Yan R, Lin W-H, Christel M, Wactlar H (2007) Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast retrieval. IEEE Trans Multimedia 9(5):958–966