Accepted Manuscript

Learning to detect video events from zero or very few video examples

Christos Tzelepis, Damianos Galanopoulos, Vasileios Mezaris, Ioannis Patras

Please cite this article as: Christos Tzelepis, Damianos Galanopoulos, Vasileios Mezaris, Ioannis Patras, Learning to detect video events from zero or very few video examples, *Image and Vision Computing* (2015), doi: 10.1016/j.imavis.2015.09.005

Learning to detect video events from zero or very few video examples

Christos Tzelepis[a,b], Damianos Galanopoulos[a], Vasileios Mezaris[a], Ioannis Patras[b]

[a]*Information Technologies Institute (ITI), CERTH, Thermi, 57001, Thessaloniki, Greece*
[b]*Queen Mary, University of London, Mile End Campus, UK, E14NS*

Abstract

In this work we deal with the problem of high-level event detection in video. Specifically, we study the challenging problems of i) learning to detect video events from solely a textual description of the event, without using any positive video examples, and ii) additionally exploiting very few positive training samples together with a small number of "related" videos. For learning only from an event's textual description, we first identify a general learning framework and then study the impact of different design choices for various stages of this framework. For additionally learning from example videos, when true positive training samples are scarce, we employ an extension of the Support Vector Machine that allows us to exploit "related" event videos by automatically introducing different weights for subsets of the videos in the overall training set. Experimental evaluations performed on the large-scale TRECVID MED 2014 video dataset provide insight on the effectiveness of the proposed methods.

Keywords: video event detection, textual event description, zero positive examples, few positive examples, related videos

Email addresses: `tzelepis@iti.gr` (Christos Tzelepis), `dgalanop@iti.gr` (Damianos Galanopoulos), `bmezaris@iti.gr` (Vasileios Mezaris), `i.patras@qmul.ac.uk` (Ioannis Patras)

1. Introduction

High-level (or complex) video event detection is the problem of finding, within a set of videos, which of them depict a given event. Typically, an event is defined as an interaction among humans or between humans and physical objects [1]. Some examples of complex events are those defined in the Multimedia Event Detection (MED) task of the TRECVID benchmarking activity [2, 3]. For instance, in MED 2014 [2], the defined complex events include *Attempting a bike trick*, *Cleaning an appliance*, or *Beekeeping*, to name a few.

The detection of such events in video has recently drawn significant attention in a wide range of applications, including video annotation and retrieval [1], video organization and summarization [4], or surveillance applications [5]. In [6], Brown studies how high-level events play a substantial role in the mechanism of structuring memories and recalling past experiences. This leads to the expectation that event-based organization of video content can significantly contribute to bridging the existing semantic gap between human and machine understanding of multimedia content.

There are several challenges associated with building an effective detector of video events. One of them is finding a video representation that reduces the gap between the traditional low-level audio-visual features that can be extracted from the video and the semantic-level actors and elementary actions that are by the definition the constituent parts of an event. In this direction, several works have shown the importance of using simpler visual concepts as a stepping stone for detecting complex events (e.g. [7, 8]). Another major challenge is to learn an association between the chosen video representation and the event or events of interest; for this, supervised machine learning methods are typically employed, together with suitably annotated training video corpora. While developing efficient and effective machine learning algorithms is a challenge in its own right, finding a sufficient number of videos that depict the event so as to use them as positive training samples for training any machine learning method is also not an easy feat. In fact, video event detection is even more challenging when

the available positive training samples are limited, or even non-existent; that is, when one needs to train an event detector using only textual information that a human can provide about the event of interest.

In this work we study the problems of i) learning an event detector solely from a textual description of the event, without using any positive video examples, and ii) learning from very few positive training samples together with a small number of "related" videos. The paper is organized as follows. In Section 2, related work in video event detection using zero or a few positive examples is reviewed. In Section 3, we present and examine different design choices for a framework that learns video event detectors based solely on textual information for training, while in Section 4 the combination of the above methods with learning from a few positive examples, as well as from related training examples, is examined. Results of the application of the proposed techniques to the TRECVID MED 2014 dataset are provided in Section 5. Finally, conclusions are drawn and discussed in Section 6.

2. Related Work

There are two broad categories concerning the learning conditions under which a video event detector is trained. That is, training may be done either by using a number of positive and negative training video examples, or using no video examples at all. Within the first category, one can distinguish between i) methods assuming that positive video examples are in abundance, and ii) methods explicitly embracing the fact that the positive samples that are available for training are typically limited, in practice. Regardless of the assumptions on the number of positive training samples, it is assumed that negative samples can be found without much effort, and thus the number of negative video samples is not a restrictive parameter in the process of learning. The above training conditions are typically simulated by the 100Ex and 10Ex MED subtasks of TRECVID [3], where 100 and 10 positive video samples are available for training video event detectors, respectively.

Training based solely on a textual description of each event class is reported in a few works, mostly in the context of the TRECVID MED 0Ex and Semantic Query (SQ) subtasks [3], where no positive video examples are provided. Instead, event detectors are trained using textual resources, which typically include the event's title, a short free-form text explanation of what may be depicted in a video that belongs to this event class, as well as brief references to visual and audio cues that are typically expected to be present in such a video (Fig. 1).

2.1. Learning from zero positive examples

Learning from zero positive examples has recently drawn significant attention in various learning problems, due to its challenging nature and the extensive applicability it has. For instance, due to the rapidly increasing number of images on the Web, extensive research efforts have been devoted in multi-label, zero-example (or few-example) classification in images [9]. Similarly, a method for zero-example classification of fMRI data was proposed in [10]. In [11], a method for predicting unseen image classes from a textual description, using knowledge transfer from textual to visual features, was proposed.

In the video domain, learning from zero positive examples is investigated primarily in the context of video event detection. In [12] this problem is addressed by transforming both the event's textual description and the visual content of un-classified videos in a high dimensional concept-based representation, using a large pool of concept detectors; then relevant videos are retrieved by computing the similarities between these representations. Similarly, in [13], each event class title is used as an input query to a text retrieval system, and the most relevant documents are retrieved. The vectorized words of these documents are then projected into the most semantically similar concepts from different modalities, such as ASR (Automatic Speech Recognition), OCR (Optical Character Recognition), and high-level features coming from applying audio-visual and DCNN (Deep Convolutional Neural Networks) concept detectors. Using these concepts, relevant videos are retrieved, and late fusion is used for combining the different

4

ranked lists of videos so as to generate the final event detection results. In [14], multiple low-level representations using both the visual and the audio content of the videos are extracted, along with higher-level semantic features coming from ASR transcripts, OCR, and off-the-shelf video concept detectors. This way, both audio-visual and textual features are expressed in a common high-dimensional concept space, where the computation of similarity is possible. In [15], logical operators are used to discover different types of composite concepts, which leads to better event detection performance.

Moreover, in [16], a relevance feedback approach is used in order to improve event detection results in the zero-example problem using features computed from several modalities. The main idea is to use the textual information that describes the event class in order to create queries for each modality. Then, the system results in ranked video lists, one per each modality. The top videos from these lists are used as a "pseudo label" video set on which a joint model is trained, and a new ranked list is produced and used for creating a new "pseudo label" set; this process is iterated a few times.

In [17], E-Lamp was proposed, which is a zero-example event detection system made of four subsystems. The first one is an off-line indexing component, while the rest of them compose the on-line event search module. In the off-line module, each video is represented with 4043 visual concepts along with ASR and OCR high-level features. Then, in the on-line search module, the user-specified event description is translated into a set of relevant concepts, called *system query*. This system query is used to retrieve the videos that are most relevant to the event. Finally, a pseudo-relevance feedback approach is exploited in order to improve the results.

Our approach goes beyond the classic semantic similarity comparison between a given event title (or other user-specified event cues) and each concept title from a concept pool. We try to enrich each concept by automatically searching in Google or Wikipedia in order to find more information for it; this enables finding semantic similarities between events and concepts more effectively. Exploiting Google Search or Wikipedia is, to the best of our knowledge,

5

novel.

2.2. Learning from a few positive examples

A limited number of studies have considered the problem of learning video event detectors from very few (e.g. 10) positive training examples [18, 19, 20, 21]. In [18], visual static (e.g. SIFT [22], Transformed Color Histogram [23]), and motion (e.g. MoSIFT [24], Improved Dense Trajectories [18]) descriptors are used, along with the Fisher Vector encoding scheme [25, 26, 27]. ASR and OCR techniques are also used for exploiting audio and textual information in video streams, as well as audio features and visual features based in DCNNs trained in ImageNet [28]. DCNN-based features typically comprise one or more of the network's hidden layers [29], providing high discriminative power [5]. Based on these features, video event detectors are trained using multiple SVM classifiers and fusion techniques (early and late) for combining different modalities. In SESAME [19], the authors also use DCNN classification scores as video features, which are subsequently fed to a kernel SVM for obtaining event detectors. ASR and low-level audio features are also extracted, and a logistic regression-based fusion technique is used for combining scores from different modalities in order to obtain the final event detectors. In [13, 30, 20], a similar training framework for learning from a few positive samples is used; low-level visual and/or audio features are combined with concept detectors' output scores for obtaining event detectors. Concept detectors are trained either using low-level features and VLAD vectors [31], or DCNN output layers.

Furthermore, it is not unusual to have in the training set videos that do not exactly fulfill the requirements to be characterized as true positive examples, but nevertheless are closely related to an event class and can be seen as "related" examples of it. This is simulated in the TRECVID MED task [3] by the "near-miss" video examples provided for each target event class. Differently from our method, none of the above works takes full advantage of these related videos for learning from few positive samples; instead, the "related" samples are either excluded from the training procedure, or they are treated as true positive or

true negative instances [32].

In our study, we will consider "related" training samples as videos that can contribute to event detector training but do not merit being treated as either true positive or true negative instances. To take advantage of them, we employ a Relevance Degree SVM (RDSVM), which can treat these samples as either weighted negatives or weighted positives in conjunction with an automatic weighting selection scheme.

3. Detecting video events using an event's textual description

3.1. General architecture

In this section we propose a framework for video event detection without using any visual knowledge about the events. We assume that the only knowledge available, with respect to each event class, is a textual description of it, which consists of a title, a free-form text, and a list of possible visual and audio cues, as in [33, 16]. Fig. 1 shows an example of such a textual description for the event class *Attempting a bike trick*. For linking this textual information with the visual content of the videos that we want to examine, similarly to [14, 18], we a) use a pool of N_c concepts along with their titles and in some cases a limited number of subtitles (e.g. concept *bicycle-built-for-two* has the subtitles *tandem bicycle* and *tandem*), and b) a pre-trained detector (based on DCNN output scores) for each concept. Fig. 2 illustrates the structure of the proposed framework. Shaded blocks indicate processing stages for which different design choices are examined in this work. Each of these stages is discussed below in detail, while Table 1 sums up the different considered design choices.

Figure 2a shows a first process that receives a textual description of an event class and a list of concepts, and generates an event detector. Given the textual description of the event class, our framework first identifies N words or phrases that most closely relate to the event class; we call this word-set the Event Language Model (ELM). In parallel, for each of the N_c concepts of our concept pool, it similarly identifies M words or phrases; we call this set the Concept Language Model (CLM) of the corresponding concept.

7

> **Event Title**: Attempting a bike trick.
>
> **Definition**: One or more people attempt to do a trick on a bicycle, motorcycle, or other type of motorized bike. To count as a bike for purposes of this event, the vehicle must have two wheels (excludes unicycles, ATVs, etc).
>
> **Visual cues:**
> - **Scene**: outdoors, often in skate park, parking lot or street.
> - **Objects/People**: person riding a bike, bike, ramps, helmet, concrete floor, audience.
> - **Activities**: riding bike on one wheel, standing on top of bike, jumping with the bike (especially over or onto objects), spinning or flipping bike.
>
> **Audio cues:**
> - **Audio**: sounds of bike hitting surface during the trick, audience cheering.

Figure 1: Textual description of the event class *Attempting a bike trick*.

Subsequently, for each word in ELM and each word in each one of CLMs we calculate the Explicit Semantic Analysis (ESA) distance [34] between them. For each CLM, the resulting $N \times M$ distance matrix expresses the relation between the given event class and the corresponding concepts. In order to compute a single score expressing this relation, we apply to this matrix different operators, such as various matrix norms or distance measures. Consequently, a score is computed for each pair of ELM and CLM. The N_c considered concepts are ordered according to these scores (in descending order) and the K top concepts along with their scores constitute our event detector. In order to perform event detection in a video collection, we compare this event detector with the output scores of concept detectors applied on each video, using different similarity measures (Fig. 2b). Thus, the final output is a ranked list of the most relevant videos. Alternatively, multiple event detectors can be generated using more than one different algorithm variations in each of the shaded blocks of Fig. 2a and 2b, and these can be used as pseudo-positive samples for training an SVM, which can then be applied to the videos so as to generate a ranked list of those depicting the target event (Fig. 2c).

3.2. Language Models

We examine the construction of three different types of ELMs, depending on the textual information that they use (Fig. 1). The first type of ELM, which will

8
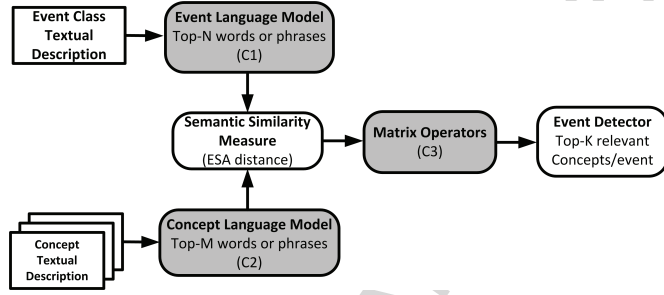
(a) Creation of event detector without positive video samples.
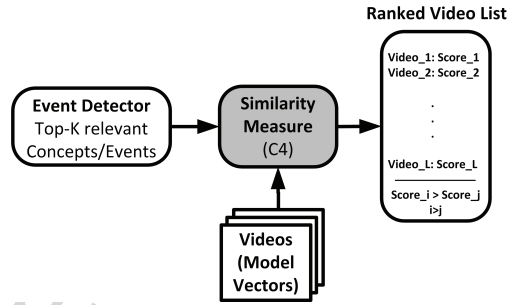
(b) Event detection in a video collection, using the event detector of the above sub-figure.
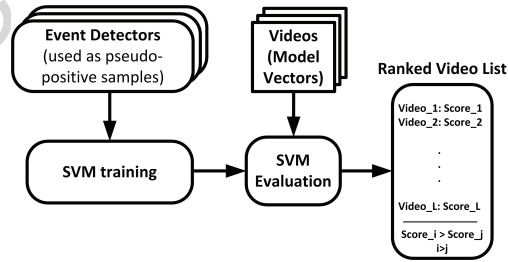
(c) Pseudo-positive sample creation and training.

Figure 2: The proposed framework for detecting video events using zero positive examples.

9

be denoted as "Title" hereafter, is based on the automatic extraction of word terms solely from the title of an event class. The second type of ELM, denoted as "Visual", is constructed by using only the visual cues provided along with the textual description of each event class. Both title and visual cues consist only of words and single phrases, such as *attempting a bike trick*, *bike*, *riding bike on one wheel*, etc. These words can automatically be extracted from the textual description without any human-expert intervention. Finally, the third type of ELM is obtained by the automatic extraction of words based on the visual and audio cues, as well as on the short free form text of an event class, and it is denoted as "AudioVisual".

Accordingly, a CLM is constructed for each one of the N_c concepts. We examine the construction of six different types of CLMs, depending on the textual information used for each concept, as well as the weighting technique (e.g. Tf-Idf) adopted for transforming this textual information in a Bag of Words (BoW) [35] representation. As a first approach, the title of a concept, along with any available subtitle, are used as a query to the Google search engine, and the text of the top-20 search results per query are retrieved. By applying text cleaning techniques (removing html tags, stop words, etc.) and Bag of Words statistics, we select the most frequently occurring words. These, together with the concept's title and subtitles, constitute the M words of the CLM. As a second approach, the concept title and any subtitle are similarly used as a query in Wikipedia, and the 20 most relevant articles per query are retrieved. By following the same procedure as above, we end up with the top-M words of the concept's CLM. As a third approach, we use only the title and the subtitles in the CLM. In the three above approaches, the Bag-of-Words (BoW) can be constructed with or without using Tf-Idf weighting [36], this resulting in six different types of CLMs.

The above types of ELMs and CLMs are introduced as different design choices, as shown in Fig. 2a. Table 1 summarizes the specific types of each one of them.

3.3. Building an event detector

The constructed language models represent the given event class and each of the available concepts as ranked lists of words. Thus, we can calculate the similarities between them by computing the semantic similarity between each word in the ELM and each word belonging to the CLMs. To this end, we use the ESA semantic relatedness measure [37], which calculates the similarity distance between two terms by computing the cosine similarity between their weighted vectors of Wikipedia articles. In this way, an $N \times M$ matrix with scores for each pair of event-concept is computed. Let S denote the aforementioned similarity matrix; that is, its (i, j)-th entry, $s_{i,j}$, denotes the similarity distance between the word w_i in ELM and the word w_j in the respective CLM. Then, we can arrive at a single score expressing the relation between the above ELM and CLM by evaluating one of the matrix operators shown in Table 1 (row C3). It is worth noting that $\lambda_{\max}(S^\top S)$ denotes the maximum eigenvalue of the covariance matrix $S^\top S$.

Processing Stage (see Fig. 2b)		Examined design choices for this processing stage
C1:	ELM	i) Title ii) Visual only iii) AudioVisual
C2a:	CLM information sources	i) Title ii) Google iii) Wikipedia
C2b:	CLM weighting	i) Tf-Idf ii) No Tf-Idf
C3:	Matrix Operation	i) ℓ_2 Norm $\quad\quad \|S\|_2 = \sqrt{\lambda_{max}(S^\top S)}$ ii) ℓ_∞ Norm $\quad\quad \|S\|_\infty = \max_{1 \le i \le N} \sum_{j=1}^{M} \lvert s_{i,j} \rvert$ iii) Frobenious Norm $\quad \|S\|_F = \left(\sum_{i=1}^{N} \sum_{j=1}^{M} \lvert s_{i,j} \rvert^2 \right)^{\frac{1}{2}}$ iv) Maximum entry $\quad \max(S)$ v) Hausdorff Distance $\quad D_{\mathcal{H}}(\mathrm{EML}, \mathrm{CML}) = \underset{p}{\mathrm{median}} \left(\max_p \|w_i - w_j\| \right)$
C4:	Distance	i) Cosine ii) Histogram Intersection iii) Kullback iv) X^2 v) Euclidean

Table 1: Design choices.

Following this process, for each event class we end up with a list of concept
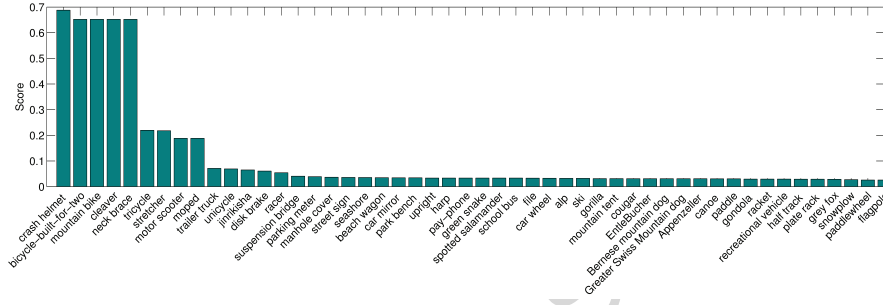
11

Figure 3: Example of event detector for the event class *Attempting a bike trick*.

scores. The event detector is then defined as the set of the top-K scores and the corresponding concept labels. An example of such an event detector is given in Fig. 3 for the event class *Attempting a bike trick*, where we observe that the three most dominant concepts are the following: *crash helmet*, *bicycle-built-for-two*, and *mountain bike*.

3.4. Applying the event detector to a video dataset

The constructed event detector is used as shown in Fig. 2b. The DCNN-based detectors for the N_c concepts in our concept pool are applied to the videos of the dataset, thus having these videos represented as vectors of concept detector output scores (hereafter called model vectors). The output scores that correspond to the K concepts comprising the event detector are selected, and are compared with the corresponding values of the event detector. For this comparison, different choices of a distance function are possible, as shown in Table 1 (row C4); that is, we examine the use of following distance functions: Euclidean, Histogram Intersection, Chi-square, Kullback-Leibler, and cosine distance. For a chosen distance function, repeating the above process for all videos in a dataset we get a ranked list, in descending order, of the videos that most closely relate to the sought event.

3.5. Using multiple event detectors as pseudo-positive training samples

Figure 2c shows an alternative approach to using event detectors that are built as presented in section 3.3. An event detector, i.e. a ranked list of concept

12

scores (Fig. 2a, Fig. 3) can be considered as a pseudo-positive sample for the respective event class. Since multiple event detectors can be obtained for a given event class by varying the design choices C1 to C3 of Fig. 2a (as shown in Table 1)), a set of pseudo-positive samples can be obtained for each event class. Negative samples for the given event class can also be obtained, in two ways. First, samples that are pseudo-positive for other event classes can be considered as pseudo-negative for a particular event class. Secondly, real videos can be selected from the Web, in analogy to how images are often selected as negative samples for training concept detectors from Web data, e.g. [38, 39]. Using these pseudo-positives and (pseudo-)negatives, a new machine learning-based event detector can be obtained by training an SVM model (for instance using the RBF kernel). This can then be evaluated on any video dataset, following the application of trained concept detectors to the videos, (as in section 3.4), to give us a new ranked list of event-related videos.

4. Learning a video event detector from a few positive and related video examples

4.1. Exploiting related videos for learning from a few examples

As discussed in section 2, in the problem of video event detection, it is not unusual to be provided, or be able to find, some related video samples, i.e. videos that are closely related with a given complex event, but do not meet the exact requirements for being a true positive event instance. Exploiting related samples can be particularly interesting when only a few true positive samples are available, since in the opposite case, when an abundance of positive samples are available, one can effectively learn from them.

In this section, Relevance Degree Support Vector Machine (RDSVM), proposed in [21] for handling "related" training samples, is employed such that related samples are taken into consideration as weighted negative or weighted positive examples, where weighting is carried out completely automatically. RDSVM extends the standard SVM algorithm such that each training sample

is assigned with a confidence value in $(0, 1]$ indicating the degree of relevance of each training sample with the class it is related.

Let $\mathcal{X} = \{(\mathbf{x}_i, y_i) \colon \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{0, \pm 1\}, i = 1, \ldots, l\}$ be an annotated dataset of l observations, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector representation of the i-th observation in the d-dimensional space with label $y_i \in \{0, \pm 1\}$ denoting that the i-th observation is a positive ($y_i = +1$), a negative ($y_i = -1$), or a related ($y_i = 0$) instance of the class. To allow the use of the RDSVM, the above is reformulated as $\mathcal{X} = \{(\mathbf{x}_i, y_i, u_i) \colon \mathbf{x}_i \in \mathbb{R}^d, y_i \in \{\pm 1\}, u_i \in \{0, 1\}, i = 1, \ldots, l\}$, where u_i is the so-called relevance label denoting that the i-th observation is a true ($u_i = 0$) or a related ($u_i = 1$) instance of the class y_i.

For the exploitation of the related observations, as proposed in [21, 40], each training sample $\mathbf{x}_i$ is associated with a adjustable confidence value v_i, and a monotonically increasing function $g(v_i)$ (called slack normalization function) is used to weight each slack variable ξ_i denoting the loss introduced by a misclassified sample. In this way, support vectors (SVs) that are associated with a higher confidence value have greater contribution to the computation of the decision function. In [21], this function is modified so that only related class observations are associated with a confidence value $c_i \in (0, 1]$ (called hereafter relevance degree) indicating the degree of relevance of the i-th observation with the class it is related. That is, g is defined as follows:

$$g(u_i) = \begin{cases} 1 & \text{if } u_i = 0, \\ c_i & \text{if } u_i = 1. \end{cases} \tag{1}$$

Thus, the contribution of the related samples in the computation of the decision function can be regulated using appropriate relevance degrees c_i.

In this study, we follow the approach proposed in [21] for handling the related samples as a subclass of the positive or negative class, for which a global relevance degree, $c = c_i \; \forall i$, is assigned to all related samples. Parameter c is optimized using a cross-validation procedure. The proposed technique for the automatic selection of related samples as weighted positives, or weighted negatives, includes the following steps. First, an RDSVM is trained using the related
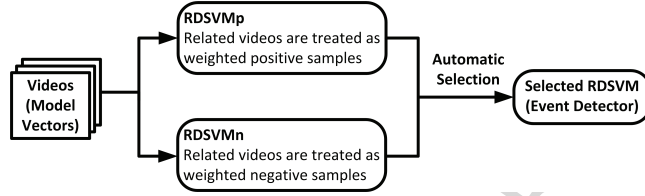
14

Figure 4: RDSVM training framework.

videos as weighted positive examples (i.e., $y_i = +1$ and $u_i = 1$). Then, another RDSVM is trained using the related videos weighted negative examples (i.e., $y_i = -1$ and $u_i = 1$). The above classifiers are denoted in Fig. 4 as RDSVM_p and RDSVM_n, respectively.

In both cases the basic parameters of the RDSVMs (e.g. C, γ for an RDSVM with RBF kernel) are optimized by conducting cross-validation (grid search) with c set to 1, i.e. treating the related samples as pure positive and pure negative samples, respectively. Subsequently, the relevance degree parameter c is optimized using cross-validation (line search) in the range $(0, 1]$. After both RDSVM_p and RDSVM_n are trained in this way, one of the two is chosen (i.e., a parameter set $\{C, \gamma, c\}$, where $c \in [-1, 1]$ is chosen) by looking at the average performance measure (e.g. average precision (AP)) values attained during cross-validation by the RDSVM_p and RDSVM_n (the one with the highest AP is selected). The chosen RDSVM is a learned event detector, that can be applied to a set of videos (where again the video representation $\mathbf{x}_i$ can be the same as in section 3: a model vector produced by application of DCNN-based concept detectors).

4.2. Treating pseudo-positive training samples as related ones

In section 3.5 we discussed how we could use the output of processing an event's textual description as a set of pseudo-positive event samples, for training a standard SVM. When we also have a few true positive videos available for training, one possibility would be to merge the two positive/pseudo-positive sets and use their union for SVM training. Since, however, the pseudo-positive samples were artificially created and thus may be corrupt by errors/noise, a better

15

option would be to treat them as "related" samples, rather than true positives. This can by straightforwardly achieved by use of the RDSVM methodology presented in the previous section. Evaluation of this approach will be presented in section 5.4.

5. Experimental results

5.1. Datasets and experimental setup

The proposed techniques are tested on the large-scale video dataset of the TRECVID Multimedia Event Detection (MED) 2014 task (hereafter referred to as MED14). The ground-truth annotated portion of it consists of three different video subsets: the "pre-specified" (PS) video subset (2000 videos, 80 hours, 20 event classes), the "ad-hoc" (AH) video subset (1000 videos, 40 hours, 10 event classes), and the "background" (BG) video subset (5000 videos, 200 hours). Each video in the above dataset belongs to either one of 30 target event classes (Table 2), or (in the case of the BG subset) to the "rest of the world" class . The above video dataset (PS+AH+BG) is partitioned such that a training and an evaluation set are created, as follows:

- **Training Set**

    – 50 positive samples per event class

    – 25 related (near-miss) samples per event class

    – 2496 background samples (negative for all event classes)

- **Evaluation Set**

    – $\sim$ 50 positive samples per event class

    – $\sim$ 25 related (near-miss) samples per event class

    – 2496 background samples (negative for all event classes)

For video representation, approximately 2 keyframes per second were extracted. Each keyframe was represented using the last hidden layer of a pre-trained Deep CNN. More specifically, a 16-layer pre-trained deep ConvNet network

16

Target Events	
E021 - Attempting a bike trick	E036 - Felling a tree
E022 - Cleaning an appliance	E037 - Parking a vehicle
E023 - Dog show	E038 - Playing fetch
E024 - Giving directions to a location	E039 - Tailgating
E025 - Marriage proposal	E040 - Tuning a musical instrument
E026 - Renovating a home	E041 - Baby Shower
E027 - Rock climbing	E042 - Building a Fire
E028 - Town hall meeting	E043 - Busking
E029 - Winning a race without a vehicle	E044 - Decorating for a Celebration
E030 - Working on a metal crafts project	E045 - Extinguishing a Fire
E031 - Beekeeping	E046 - Making a Purchase
E032 - Wedding shower	E047 - Modeling
E033 - Non-motorized vehicle repair	E048 - Doing a Magic Trick
E034 - Fixing musical instrument	E049 - Putting on Additional Apparel
E035 - Horse riding competition	E050 - Teaching Dance Choreography

Table 2: Target events in the employed TRECVID MED dataset.

provided in [29] was used. This network had been trained on the ImageNet data [28] and provides scores for 1000 concepts. Thus, each keyframe has a 1000-element vector representation. Then, a video-level model vector for each video is computed by taking the average of the corresponding keyframe-level representations.

5.2. Video event detection using the event's textual description

The framework proposed in Section 3 allows for different design choices in its various stages (Table 1, and shaded processing stages in Fig. 2). There are 5 stages that can be parameterized: the ELM and CLM information sources selection (C1, C2a), the CLM textual vector weighting strategy (C2b), the matrix operator used (C3), and the distance function selected (C4). Based on the possible choices listed in Table 1, 450 different combinations are possible and were tested.

Table 3 presents the 10 best-performing combinations in terms of mean average precision (MAP) across all 30 events. Note that, when the "Title" is used for both ELM and CLM construction (processing stages C1 and C2a), meaning that both ELM and CLMs are represented by a single word or phrase ($N = M = 1$), then all matrix operations result in the same score, making no difference in the final result. This is the reason why a dash sometimes appears in the C3

17

column of Table 3. Moreover, when "Title" is selected for the construction of the CLMs (C2a processing stage), there is no need for Bag-of-Words encoding, thus no use of weighting technique (Tf-Idf), since no enrichment by searching in Google or in Wikipedia was carried out. This is why there is no choice in the weighting stage (C2b) when "Title" is selected. As can be seen, the enrichment of the CLM through Google search, without Tf-Idf weighting and the usage of as much as possible information for constructing the EML resulted in the best result overall.

In order to explore the effectiveness of the different design choice combinations, we experimented with changing one design choice at a time, keeping the choices for all other processing stages unaffected (as in the best-performing combination of Table 3). Table 4 shows the performance of different ELMs in stage C1, clearly suggesting that the exploitation of as much as possible information for the particular parameter leads to better detection results. In Table 5, it is shown that enrichment of the CLM in stage C2a by searching in Google is the most effective way to do so. As Table 6 shows, using Tf-Idf weighting does not seem to provide any improvement to the detection performance. Table 7 suggests that the use of the Hausdorff distance, as a similarity matrix operator, outperforms the rest of the respective choices. Finally, Table 8 shows the performance of different similarity measures (stage C4) in the event detection step (Fig. 2b), where the cosine and the Histogram Intersection measures outperform the rest of them.

In Table 9, we present the performance of the training step of our framework, as illustrated in Fig. 2c (denoted as T_{10}), compared to the best combination of the event detection step, shown in Fig. 2b (denoted as T_0). As discussed above, for T_{10} we consider the event detectors generated as shown in Fig. 2a as pseudo-positive instances of the respective event class. Also, as discussed previously, negative samples can be obtained using two different approaches. First, the pseudo-positive samples from the rest of the event classes are used as pseudo-negative samples, and, secondly, real negative videos for all event classes, belonging to the "background" (BG) training subset, are used. Finally, a linear

18

late fusion approach, using the arithmetic mean operator, is used in order to improve the individual detection results, denoted as $T_0 \oplus T_{10}$. Hereafter, we will denote with $\oplus$ the late fusion scheme where the arithmetic mean operator is used for combining the event detection output scores. We can observe that the combination of the event detection stage, T_0, (Fig. 2b) with SVM training with pseudo-negative samples, T_{10}, (Fig. 2c) achieves the best performance, resulting in MAP equal to 12.38%.

C1 (ELM)	C2a (CLM)	C2b (Weighting)	C3 (Matrix Operation)	C4 (Distance)	MAP
AudioVisual	Google	No Tf-Idf	Hausdorff	Cosine	0.1111
AudioVisual	Google	No Tf-Idf	Hausdorff	Histog_Inter	0.1109
AudioVisual	Google	No Tf-Idf	Hausdorff	Kullback	0.1054
Title	Title	-	-	Histog_Inter	0.1045
Visual	Google	No Tf-Idf	Hausdorff	Cosine	0.1005
Visual	Google	No Tf-Idf	Hausdorff	Histog_Inter	0.0991
Title	Title	-	-	Cosine	0.0988
AudioVisual	Google	Tf-Idf	Hausdorff	Histog_Inter	0.0978
Visual	Google	No Tf-Idf	Hausdorff	Kullback	0.0956
AudioVisual	Google	Tf-Idf	Hausdorff	Cosine	0.0933

Table 3: Top-10 parameter combinations in terms of MAP.

C1 (ELM)	C2a (CLM)	C2b (Weighting)	C3 (Matrix Operation)	C4 (Distance)	MAP
AudioVisual					0.1111
Visual	Google	No Tf-Idf	Hausdorff	Cosine	0.1005
Title					0.0760

Table 4: Comparison of different Event Language Models (ELMs).

C1 (ELM)	C2a (CLM)	C2b (Weighting)	C3 (Matrix Operation)	C4 (Distance)	MAP
	Google				0.1111
AudioVisual	Wikipedia	No Tf-Idf	Hausdorff	Cosine	0.0850
	Title				0.0760

Table 5: Comparison of different Concept Language Models (CLMs).

We compare the proposed method with the E-Lamp framework, a state-of-the-art system that participated in the TRECVID 2014 MED task [18] and is described in detail in [17]. As mentioned in section 2.1, the E-Lamp system consists of four major subsystems, namely Video Semantic Indexing (VSIN), Semantic Query Generator (SQG), Multimodal Search (MS) and Pseudo-Relevance

C1 (ELM)	C2a (CLM)	C2b (Weighting)	C3 (Matrix Operation)	C4 (Distance)	MAP
AudioVisual	Google	No Tf-Idf Tf-Idf	Hausdorff	Cosine	0.1111 0.1005

Table 6: Comparison of different Weighting schemes.

C1 (ELM)	C2a (CLM)	C2b (Weighting)	C3 (Matrix Operation)	C4 (Distance)	MAP
AudioVisual	Google	No Tf-Idf	Hausdorff l_2 Frobenius l_∞ Max	Cosine	0.1111 0.0833 0.0827 0.0828 0.0804

Table 7: Comparison of various norms of the $N \times M$ distance matrix.

Feedback (PRF). The VSIN subsystem represents the input videos as a set of low- and high-level features from several modalities. The high-level features, i.e. the result of semantic concept detection, are used as input to the SQG subsystem, in which the textual description of an event class is translated into a set of relevant concepts termed *system query*. The system query is then used in the MS subsystem as input to several well-known text retrieval models in order to find the most relevant videos. These results can be then refined by the PRF subsystem.

As SQG leads to the creation of an event detector using semantic concepts, a correspondence exists (and comparison is possible) with our approach to build an event detector as described in sections 3.2 and 3.3 (Fig. 2a). Similarly, the MS subsystem corresponds to (and can be compared with) our event detection module presented in section 3.4 (Fig. 2b). We compared with four SQG approaches that are presented in the E-Lamp system. These are: i) Exact word matching, ii) WordNet mapping using Wu & Palmer measure (Wu) iii) WordNet mapping using the structural depth distance in WordNet hierarchy (Path), and iv) Word Embedding Mapping (WEP). Concerning the MS stage, we compared with the following retrieval methods: the Vector Space Model [33], the Okapi BM25, and two unigram language models with Jelinek-Mercer smoothing (LM-JM) and Dirichlet smoothing (LM-DL) respectively [41].

Table 10 shows the performance, in terms of mean average precision (MAP),

ELM	CLM	Weighting	Matrix Operation	Distance	MAP
				Cosine	0.1111
				Histog_Inter	0.1109
AudioVisual	Google	No Tf-Idf	Hausdorff	Kullback	0.1054
				X^2	0.0832
				Euclidean	0.0690

Table 8: Comparison of different similarities measures.

Experimental Scenario	T_0	T_{10} Pseudo-Negatives	T_{10} Real Negatives	$T_0 \oplus T_{10}$
MAP	0.1111	0.10125	0.0594	**0.1238**

Table 9: The performance of the developed techniques in AP, and across 30 events in MAP.

of the above combinations in comparison to the best-performing event detector and distances proposed in the present work (Table 8). From this Table it is clear that the proposed method for building an event detector outperforms the rest of the compared methods, irrespective of the similarity measure that they are combined with. Out of the event detection creation methods of [17], the *exact word* seems to perform considerably better than the others (but much worse than the proposed method). This is because the concept labels from our concept pool that are most related to an event are often well-represented in the event's textual description, e.g. for the event *Beekeeping*, the word bee is observed 31 times, and this word is directly associated with the concepts *bee* and *bee eater*. The WordNet and WEP mappings on the other hand, are not always successful in finding the semantic similarity between two words. Regarding the compared similarity measures, the VSM and LM-DL generally perform better than BM25 and LM-JM, but the proposed cosine and Histogram Intersection distances are consistently among the top-performing measures. It should be noted that the number of visual concepts used in [17] is significantly greater than the 1000 concepts used throughout our experiments, and other modalities (e.g. audio) are also exploited in [17]; this explains the often higher MAP values that are reported in the latter work.

21

Event detector creation	Similarity measures [17]				Similarity measures (proposed)	
	VSM	BM25	LM-JM	LM-DL	Cosine	Histogram Intersection
WordNet - Wu	0.0205	0.0201	0.0250	0.0319	0.0222	0.0318
WordNet - Path	0.0333	0.0221	0.0310	0.0379	0.0359	0.0434
Exact word	0.0833	0.0287	0.0541	0.0568	0.0828	0.0801
WEM	0.0429	0.0232	0.0269	0.0331	0.0427	0.0418
Proposed	0.0912	0.0980	0.0392	0.0993	**0.1111**	0.1109

Table 10: Comparison between proposed and compared methods.

5.3. Learning from a few positive and related video examples

In this section, we validate the performance of the proposed framework for handling related samples for the problem of learning video event detectors from a few positive samples. To this end, we compare the following approaches in order to investigate whether and in what way it is beneficial to use related samples in training: a) using no related samples, b) related samples are used as pure negative ones, as in [32], c) related samples are used as pure positive ones, again as in [32], and d) related samples are used as weighted negative or positive ones under the RD-SVM framework of Section 4.1, which involves an automatic procedure for selecting both the labels of the related samples and their weights.

As discussed in section 5.1, the MED14 dataset provides 50 positive samples per each of the 30 event classes. However, we want to simulate the case where only a few positive samples are available for training, thus we choose to use only 10 training samples per event class. To this end, for each experiment we randomly draw a subset of 10 positive samples for each event class, and we repeat this 10 times. That is, for each compared approach in this section, the obtained performance of the corresponding classifier (RD-SVM or standard SVM) is averaged over 10 iterations. For the approaches that use related samples, 10 such samples were chosen from the pool of 25 related samples that are available for each event class; these, as suggested in [21], were selected as the 10 that are the nearest to the median of all 25 related samples in the employed feature space.

Table 11 shows the results of these comparisons, in terms of mean average precision (MAP), across the 30 event classes of the MED14 dataset. We observe

22

that when related samples are used as pure negative or pure positive ones as in [32], the overall detection performance is lower than the baseline approach which does not use these samples at all (P_{10}). In contrast to this, the proposed approach that treats related samples as positive/negative ones with automatic weighting achieved better performance than the baseline, reaching a MAP of 18.95%.

In Fig. 5 the results of the above comparisons are given separately for each of the 30 event classes of the dataset. As can be seen, despite the fact that treating related samples as either pure negatives or pure positives jointly for all the event classes leads to worse average detection performance (MAP), compared to the case where they are not used at all in the training process, there are event classes where it is beneficial to use them in such a way. Specifically, for 15 out of 30 events it is better to use related samples as pure positives rather that to exclude them from the training process, and similarly for one event it is beneficial to use them as pure negatives. Automatically selecting to use related samples as weighted negatives or weighted positives using RD-SVM [21], on the other hand, is better than excluding them from the annotation process for 19 out of the 30 event classes. Moreover, the automatic weight selection leads to better results than both approaches that use related samples without any weight for 18 out of the 30 events. The above confirm our hypothesis that treating related samples as weighted negatives/positives using RD-SVM [21] can lead to better event detection performance.

Experimental Scenario	P_{10} (baseline)	Related as negatives (as in [32])	Related as positives (as in [32])	R_{10} (proposed)
MAP	0.1808	0.1368	0.1796	**0.1895**

Table 11: Comparisons of approaches using related samples for learning an event detector.

5.4. Combining textual event detectors with learning from a few positive and related samples

Our first attempt to combine textual event detectors and learning from a few examples is based on exploiting the pseudo-positive samples, which were
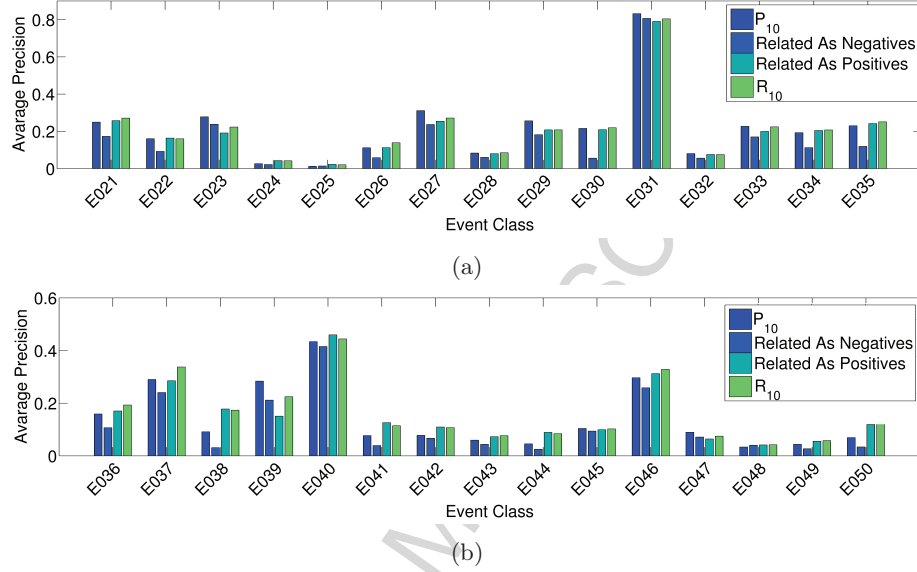
Figure 5: Experimental evaluation of various methods for treating related samples for event classes (a) E021-E035 and (b) E036-E050.

computed according to section 3.5, as related samples in RD-SVM. We chose a subset of 10 pseudo-positive samples for each event class, using a same selection strategy as in the case of related samples in section 5.3; that is, the 10 nearest to the median pseudo-positive sample for each event class were selected. Using RDSVM for handling pseudo-positive samples as weighted negative or positive ones (with automatic weighting selection) resulted in a MAP equal to 18.26%, across 30 events (denoted as R_{10p} in Table 12). It is worth noting that, similarly to the previous results, this experimental result stands for the average of 10 iterations (using 10 different, randomly selected sets of 10 positive examples each). We observe that, using pseudo-positive samples this way outperforms learning from solely 10 positive samples by a small margin.

In a second attempt to further combine text-based and learning-based event detectors, we examine the late fusion of detectors. As shown in the last column of Table 9 , $T_0 \oplus T_{10}$ denotes the best approach for learning video event detectors using the textual description of each event class alone, resulting in a MAP equal to 12.38%. The results of combining the above approach, as well as R_{10p} (which
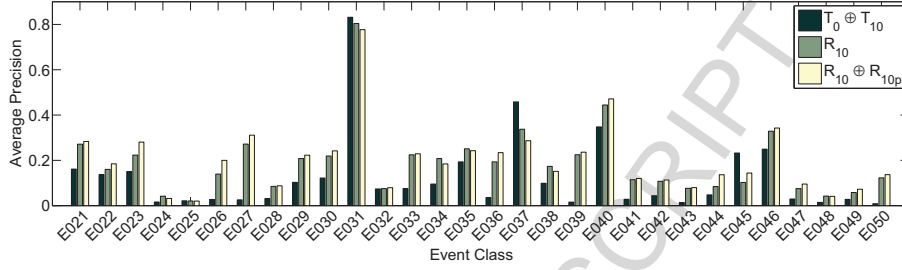
24

Figure 6: Event detection performance (AP) for 30 event classes of MED14 dataset using three different fusion approaches.

already jointly uses textual information and real training samples) with the P_{10} and R_{10} learned detectors (Table 11), are shown in Table 12. As we can see, the best-performing combination is that of R_{10} and R_{10p}, which achieves MAP equal to 20.11%. This is higher that the best performance that is achieved in our experiments using visual examples alone (R_{10}), and much higher than that achieved using only textual information about the sought event, highlighting the importance of combining visual examples and a textual description of the event for learning. Other combinations, most notably that of $T_0 \oplus T_{10}$ with P_{10}, R_{10} or R_{10p}, do not seem to offer any improvement over R_{10p} alone. This can be attributed to the fact that the $T_0 \oplus T_{10}$ detector is much weaker than P_{10}, R_{10}, and R_{10p}, thus introducing mostly noise to the results of the latter at the late fusion stage.

Combinations of event detectors	MAP
R_{10p}	0.1826
$T_0 \oplus T_{10} \oplus P_{10}$	0.1743
$T_0 \oplus T_{10} \oplus R_{10}$	0.1662
$T_0 \oplus T_{10} \oplus R_{10p}$	0.1647
$P_{10} \oplus R_{10}$	0.1893
$P_{10} \oplus R_{10p}$	0.1971
$P_{10} \oplus R_{10} \oplus R_{10p}$	0.1965
$R_{10} \oplus R_{10p}$	**0.2011**

Table 12: Various combinations of textual event detectors with learning from few positive, related, and pseudo-positive training samples.

In Fig. 6, we present the event detection results per event class, using Average Precision (AP), for i) the best-performing proposed approach that learns from

25

the events' textual descriptions (last column of Table 9); ii) the best-performing proposed approach for learning from a few positive and related examples (last column of Table 11); and, iii) the best combination of Table 12 (last row) for exploiting both video examples and a textual description of the event. We observe that the latter combination outperforms the former two approaches for 22 of the 30 event classes. Notable exceptions, where using just the textual description of the event performs the best, are events E031 (*Beekeeping*), E037 (*Parking a vehicle*), and E045 (*Extinguishing a Fire*). This can be attributed to the fact that we have in our concept pool a wealth of concepts related to these events (e.g. for E037: *beach wagon, car mirror, electric locomotive, minibus, parking meter, recreational vehicle, sports car, streetcar, etc.*; for E045: *fire engine, fire screen, fireboat, gas pump, stove, water tower, etc.*) and at the same time the textual description which is performed for these events allows us to identify those related concepts (e.g. for E031, the top-4 concepts that $T_0 \oplus T_{10}$ identifies are: *honeycomb, apiary, bee, bee eater*). In contrast to this, for several of other events only one or two concept (or even no concepts at all) closely relate to them, e.g. for event E047 (*Modeling*) the closest-related concept that is included in our concept pool is *kimono* and, similarly, concept *grocery store* for event E046 *Making a Purchase*.

6. Conclusion

In this paper we proposed a framework for learning video event detectors from solely a textual description of an event class, or from a very few positive and related training samples. We identified a general learning framework and studied the impact of various design choices for different stages of this framework. For exploiting related video samples we employed an SVM extension (RDSVM) such that related samples are automatically treated as weighted negative or positive samples. The experimental evaluation of the proposed approaches, as well as the combination of them on the challenging, large-scale TRECVID MED 2014 video dataset verified the applicability of the proposed methods in the cases

where positive samples are not available, or scarce, and provided useful insight on how to train video event detectors under such conditions.

7. Acknowledgment

References

[1] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, M. Shah, High-level event recognition in unconstrained videos, International Journal of Multimedia Information Retrieval (2012) 1–29.

[2] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, G. Quenot, An overview of the goals, tasks, data, evaluation mechanisms and metrics, in: Proc. of TRECVID 2013, NIST, USA, 2013.

[3] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, G. Quenot, An overview of the goals, tasks, data, evaluation mechanisms and metrics, in: Proc. of TRECVID 2014, NIST, USA, 2014.

[4] D. Zhang, S.-F. Chang, Event detection in baseball video using super-imposed caption recognition, in: Proc. of the 10th ACM Int. Conf. on Multimedia, ACM, 2002, pp. 315–318.

[5] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, et al., A large-scale benchmark dataset for event recognition in surveillance video, in: Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on, IEEE, 2011, pp. 3153–3160.

[6] N. R. Brown, On the prevalence of event clusters in autobiographical memory, Social Cognition 23 (2005) 35–69.

[7] N. Gkalelis, V. Mezaris, M. Dimopoulos, I. Kompatsiaris, T. Stathaki, Video event detection using a subclass recoding error-correcting output

codes framework, in: Multimedia and Expo (ICME), IEEE Int. Conf. on, IEEE, 2013, pp. 1–6.

[8] N. Gkalelis, V. Mezaris, Video event detection using generalized subclass discriminant analysis and linear support vector machines, in: Proc. of Int. Conf. on multimedia retrieval, ACM, 2014, p. 25.

[9] T. Mensink, E. Gavves, C. G. Snoek, COSTA: Co-occurrence statistics for zero-shot classification, in: Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on, IEEE, 2014, pp. 2441–2448.

[10] M. Palatucci, D. Pomerleau, G. E. Hinton, T. M. Mitchell, Zero-shot learning with semantic output codes, in: Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, A. Culotta (Eds.), Advances in Neural Information Processing Systems 22, Curran Associates, Inc., 2009, pp. 1410–1418.

[11] M. Elhoseiny, B. Saleh, A. Elgammal, Write a classifier: Zero-shot learning using purely textual descriptions, in: Computer Vision (ICCV), IEEE Int. Conf. on, IEEE, 2013, pp. 2584–2591.

[12] A. Habibian, T. Mensink, C. G. Snoek, Videostory: A new multimedia embedding for few-example recognition and translation of events, in: Proc. of the ACM Int. Conf. on Multimedia, ACM, 2014, pp. 17–26.

[13] Y. Guangnan, L. Dong, C. Shih-Fu, S. Ruslan, M. Vlad, D. Larry, G. Abhinav, H. Ismail, G. Sadiye, M. Ashutosh, BBN VISER TRECVID 2014 multimedia event detection and multimedia event recounting systems, in: Proc. TRECVID Workshop, 2014.

[14] S. Wu, S. Bondugula, F. Luisier, X. Zhuang, P. Natarajan, Zero-shot event detection using multi-modal fusion of weakly supervised concepts, in: Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on, IEEE, 2014, pp. 2665–2672.

[15] A. Habibian, T. Mensink, C. G. Snoek, Composite concept discovery for zero-shot video event detection, in: Proc. of Int. Conf. on Multimedia Retrieval, ACM, 2014, p. 17.

[16] L. Jiang, T. Mitamura, S.-I. Yu, A. G. Hauptmann, Zero-example event search using multimodal pseudo relevance feedback, in: Proc. of Int. Conf. on Multimedia Retrieval, ACM, 2014, p. 297.

[17] L. Jiang, S.-I. Yu, D. Meng, T. Mitamura, A. G. Hauptmann, Bridging the ultimate semantic gap: A semantic search engine for internet videos, in: Int. Conf. on Multimedia Retrieval, 2015.

[18] S.-I. Yu, L. Jiang, Z. Mao, X. Chang, X. Du, C. Gan, Z. Lan, Z. Xu, X. Li, Y. Cai, et al., Informedia at TRECVID 2014 MED and MER, in: NIST TRECVID Video Retrieval Evaluation Workshop, 2014.

[19] R. Bolles, B. Burns, J. Herson, et al., The 2014 SESAME multimedia event detection and recounting system, in: Proc. TRECVID Workshop, 2014.

[20] N. Gkalelis, F. Markatopoulou, A. Moumtzidou, D. Galanopoulos, K. Avgerinakis, N. Pittaras, S. Vrochidis, V. Mezaris, I. Kompatsiaris, I. Patras, ITI-CERTH participation to TRECVID 2014, in: Proc. TRECVID Workshop, 2014.

[21] C. Tzelepis, N. Gkalelis, V. Mezaris, I. Kompatsiaris, Improving event detection using related videos and relevance degree support vector machines, in: Proceedings of the 21st ACM Int. Conf. on Multimedia, ACM, 2013, pp. 673–676.

[22] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[23] K. E. Van De Sande, T. Gevers, C. G. Snoek, Evaluating color descriptors for object and scene recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 32 (9) (2010) 1582–1596.

29

[24] M.-y. Chen, A. Hauptmann, MoSIFT: Recognizing human actions in surveillance videos, Technical Report.

[25] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: Theory and practice, International Journal of Computer Vision 105 (3) (2013) 222–245.

[26] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: Computer Vision–ECCV 2010, Springer, 2010, pp. 143–156.

[27] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: British Machine Vision Conference, 2011.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition (CVPR), IEEE Conf. on, IEEE, 2009, pp. 248–255.

[29] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[30] H. Cheng, J. Liu, I. Chakraborty, G. Chen, Q. Liu, M. Elhoseiny, G. Gan, A. Divakaran, H. Sawhney, J. Allan, J. Foley, M. Shah, A. Dehghan, M. Witbrock, J. Curtis, SRI-Sarnoff AURORA system at TRECVID 2014 multimedia event detection and recounting, in: Proc. TRECVID Workshop, 2014.

[31] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (9) (2012) 1704–1716.

[32] M. Douze, D. Oneata, M. Paulin, C. Leray, N. Chesneau, D. Potapov, J. Verbeek, K. Alahari, Z. Harchaoui, L. Lamel, J.-L. Gauvain, C. A. Schmidt, C. Schmid, The INRIA-LIM-VocR and AXES submissions to TRECVID 2014 multimedia event detection.

[33] E. Younessian, T. Mitamura, A. Hauptmann, Multimodal knowledge-based analysis in multimedia event detection, in: Proc. of the 2nd ACM Int. Conf. on Multimedia Retrieval, ACM, 2012, pp. 51:1–51:8.

[34] E. Gabrilovich, S. Markovitch, Computing semantic relatedness using wikipedia-based explicit semantic analysis., in: IJCAI, Vol. 7, 2007, pp. 1606–1611.

[35] G. Salton, M. J. McGill, Introduction to modern information retrieval, McGraw-Hill, Inc., 1986.

[36] J. Leskovec, A. Rajaraman, J. D. Ullman, Mining of massive datasets, Cambridge University Press, 2014.

[37] D. Carvalho, C. Callı, A. Freitas, E. Curry, EasyESA: A low-effort infrastructure for explicit semantic analysis, in: Proc. of the 13th Int. Semantic Web Conf. (ISWC), 2014, pp. 177–180.

[38] R. Yan, A. G. Hauptmann, R. Jin, Negative pseudo-relevance feedback in content-based video retrieval, in: Proc. of the 11th ACM Int. Conf. on Multimedia, ACM, 2003, pp. 343–346.

[39] X. Li, C. G. Snoek, M. Worring, D. Koelma, A. W. Smeulders, Bootstrapping visual categorization with relevant negatives, IEEE Transactions on Multimedia 15 (4) (2013) 933–945.

[40] X. Wu, R. Srihari, Incorporating prior knowledge with weighted margin support vector machines, in: Proc. 10th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining, ACM, 2004, pp. 326–333.

[41] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to information retrieval, ACM Transactions on Information Systems (TOIS) 22 (2) (2004) 179–214.

Highlights

- We deal with the challenging problem of high-level event detection in video.
- We build event detectors based solely on textual descriptions of the event classes.
- We also learn event detectors from very few positive and related training samples.
- We present results and comparisons on a large-scale TRECVID MED video dataset.