Learning from Web Videos for Event Classification

Nicolas Chesneau, Karteek Alahari, *Senior Member, IEEE* and Cordelia Schmid, *Fellow, IEEE*

***Abstract*—Traditional approaches for classifying event videos rely on a manually curated training dataset. While this paradigm has achieved excellent results on benchmarks such as TrecVid multimedia event detection (MED) challenge datasets, it is restricted by the effort involved in careful annotation. Recent approaches have attempted to address the need for annotation by automatically extracting images from the web, or generating queries to retrieve videos. In the former case, they fail to exploit additional cues provided by video data, while in the latter, they still require some manual annotation to generate relevant queries. We take an alternate approach in this paper, leveraging the synergy between visual video data and the associated textual metadata, to learn event classifiers without manually annotating any videos. Specifically, we first collect a video dataset with queries constructed automatically from textual description of events, prune irrelevant videos with text and video data, and then learn the corresponding event classifiers. We evaluate this approach in the challenging setting where no manually annotated training set is available, i.e., EK0 in the TrecVid challenge, and show state-of-the-art results on MED 2011 and 2013 datasets.**

***Index Terms*—Event classification, Convolutional neural networks, Self-supervised learning.**

I. Introduction

THE problem of classifying complex events has become ever more challenging in the context of datasets comprising videos from diverse sources. This task is highlighted by competitions such as the TrecVid multimedia event detection (MED) challenge, which is being held annually since 2010 [33]. The standard approach to address this classification problem is to extract features from video, learn a classifier for each event with a training dataset of videos, and then evaluate it on the test set [8], [18], [19], [21], [22], [24], [35], [42], [45]. While methods in this paradigm vary in terms of feature representation, from spatio-temporal or volumetric models [19], [21], [22], [24] to dense trajectories [42], and then to features learned with convolutional neural networks [8], [18], [35], [45], they all rely on manually annotated training videos. This makes it difficult to scale them up to data collections with a large number of classes, given the lack of reliable, sufficiently large public training sets. In the past few years, several approaches have been proposed to overcome the need for fully-annotated training data for event classification [4], [6], [13], [31], [32], [37]. Some of these methods build a training set incrementally, by first learning a classifier from an initial dataset and then using it to retrieve additional samples from the web [13]. An alternative to this strategy is to learn multiple classifiers, and combine them with learned weights [4] or a multiple instance learning (MIL) framework [32]. Although these approaches showed interesting results, they do not exploit the rich cues present in metadata (in the form of text) associated with image or video content on the web [6], [13], [32] or are limited to using only image data [37]. This paper focuses on addressing such limitations of zero-example event classification methods, wherein no manually-curated video training data is available to learn the models, see Figure 1.

Birthday party
An individual celebrates a birthday with other people.
A birthday in this context is the anniversary of a person's birth. Less commonly, the term "birthday" can be used to refer to theanniversary of an organization's establishment, but a celebration for anorganization does not satisfy the event definition. A birthday celebration is a gathering of people who have been invited bythe host or hosts to come to a set location (often a private home,sometimes a restaurant, bar, nightclub, park, or other public venue) tosocialize in honor of the person(s) whose birthday it is (the birthdaycelebrant(s)). Birthday parties, as with other parties/celebrations, will typicallyfeature an assortment of food and beverages. Birthday parties are oftenaccompanied by colorful decorations, such as balloons and streamers, andsome people may wear cone-shaped "birthday hats". The decorations mayinclude signs or banners displaying a message for the birthdaycelebrant. Often, especially for children's parties, guests will bringcards and/or gifts wrapped in shiny/colorful paper or bags, which will beopened by the birthday celebrant(s), or by their parents/siblings if thebirthday celebrants are too young to open the gifts by themselves. A cake (or sometimes cupcake or other food item) with lit candles, called the"birthday cake", is often served. The song "Happy Birthday to You" may sungby the guests while the birthday cake with lit candles is carried to atable or counter where the birthday celebrant(s) are seated. The birthdaycelebrant(s) then blow out the candles, usually after the song is finished,and the guests then clap and cheer. Birthday parties may also involve gamesor other organized group activities.

Evidential description: scene: indoors (a home, a restaurant) or outdoors (backyard, park); day or night objects/people: decorations (balloons, streamers, conical hats, etc), birthday cake (often with candles), birthday celebrant, guests, gifts activities: singing, blowing out candles on cake,

Fig. 1. Given a textual description of a category (left), here *birthday party*, the goal is to rank a set of (test) videos to find relevant videos of the category (right). Our goal in this paper is to learn a classifier without any manually annotated training videos.

The core of the proposed approach is the synergy between text and visual content. It begins by analyzing the given textual description of each event, which consists of event name (*Birthday party* in Figure 1), a one-phrase definition ("An individual celebrates birthday with other people.") and a short description ("A birthday in this context...") [34], to automatically extract a set of queries (see Section III-A). To this end, we use natural language processing techniques to extract keywords relevant to the event, and perform query expansion to further enrich the initial query based on the event name. We then query YouTube to collect an initial training set. This set is automatically pruned, with our novel algorithm, using text and vision-based features to retain only the most relevant video content (see Section III-B). Each selected video is then represented with state-of-the-art convolutional neural network (CNN) features [20], [36], together with dense trajectories [42], to learn event classifiers (see Section III-C). We analyze the impact of the different steps in our algorithm, namely, query generation, expansion, and pruning on the TrecVid 2011 test set of 31,820 videos, and then compare to state-of-the-art methods [15], [16], [37] on the TrecVid MED 2013 EK0 dataset [34], which contains 24,957 test videos. We show that our method achieves the best performance, with a significant improvement of more than 30% mean average precision (mAP) over recent results [15] on this dataset (see Section IV).

The authors are with Inria (Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France).

II. Related Work

The traditional setup for event classification is where a dataset of labeled videos is provided to train models [8], [18], [19], [21], [22], [24], [35], [42], [45]. Such methods, based on a myriad of features, have achieved excellent results in several TrecVid MED challenges over the years. Our work in this paper also focuses on the event classification problem, but in the much more challenging setting—when no training video dataset is provided—such as the TrecVid MED EK0[1] challenges.

Several innovative approaches have been proposed to address the availability of limited or no (zero-example) training data for event classification. Niebles *et al.* [31] represented events, in particular those performed by humans, with topic models, which were learned with probabilistic latent semantic analysis and latent Dirichlet allocation. This framework, was however, evaluated on a limited set of sequences, and it is unclear if it would generalize well to the unconstrained setting we consider in this paper. Other approaches like [4], [6], [7], [13], [30], [32] have used web resources to collect a training set. For example, [4], [13], [32] learn a classifier with an initial training set, and then use it to collect additional samples from the web, with Google, Bing and YouTube search. Duan *et al.* [6] proposed a transfer learning scheme on videos collected from YouTube. Although these approaches have made notable progress, they fall short in one or more of the following ways: (i) lack of ability to exploit rich cues present in textual description of events [6], [7], [13], [30], [32], (ii) reliance on some form of manual annotation [4], [6], [32], (iii) limited to using image data [7], [13], [37]. We address these limitations in this paper with an approach exploiting additional cues in text metadata, in the challenging setting where no manual annotation is available for large TrecVid datasets, i.e., TrecVid EK0.

One way to address some of the limitations discussed above is by using text as additional information [23], [39], [43], [46], inspired by early methods for video segmentation [11] and video summarization [38]. Such techniques were seldom deployed on a large scale, and were ahead of their time. Song *et al.* [39] adapt classifiers learned on labeled text documents to videos, by treating them as weak classifiers in a boosting framework. A strong video classifier is then learned by combining the weak responses with a classifier trained on labeled videos. Another boosting approach [23] combined text metadata and video feature classifiers in a MIL framework, but relied on a training set of videos, annotated by expert and amateur human labelers. Similarly, the method in [43] requires a manually curated initial training set to extract additional data from webpages and related videos. While these methods demonstrated the benefits of using text, they still require video annotations, unlike our method, where none of the video data is manually labeled.

An alternative way to use text in combination with visual features is to define a set of concepts, with individual words or short phrases, that describe events, and are simpler to learn. Works such as [3], [7], [9], [25] learn a model to detect events with such concepts. Improvements to this scheme include modeling a pair of words or n-grams to not only exploit the co-occurrences between words [27], but also disambiguate among the multiple meanings represented by individual words [5]. Works in this paradigm, where events are represented as a collection of concept responses, are increasingly leveraging weakly annotated data from the internet [3], [44]. The method in [46] builds a list of frequent words in the text metadata, which represent concepts, to prune videos downloaded from YouTube. Singh *et al.* [37] present a similar approach, where an initial set of concepts is extracted from the textual description of an event, which is then pruned to ultimately obtain an image dataset for training visual concepts. Despite promising results, this method solely relies on images to differentiate between events. Consider two example events from the TrecVid event retrieval challenge: *changing vehicle tire* and *unstuck vehicle*, see Figure 2. It is nearly impossible to distinguish between these events simply from images. Jiang *et al.* [15], [16] use video data instead of images, and a self-paced learning scheme to train concept detectors, but still require a small set of reliably-annotated video samples. In contrast, our method automatically curates a video training set. In Section IV we empirically compare to all these related approaches, and clearly show the benefits of our framework.

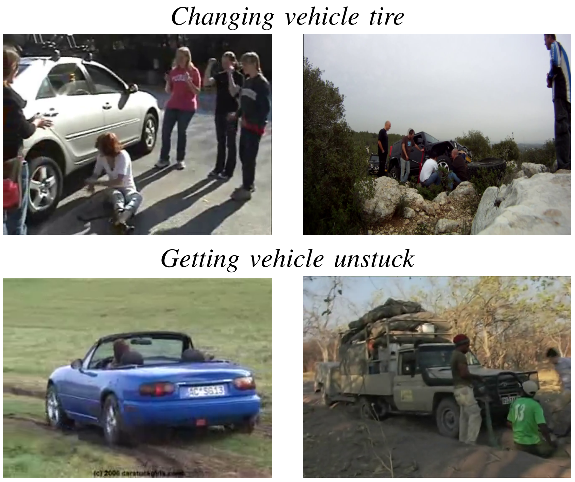


Fig. 2. Sample frames from two events: *changing vehicle tire* (top) and *unstuck vehicle* (bottom). These events are nearly impossible to distinguish from images alone, and cannot be handled effectively by previous methods [37].

III. Learning from Web Videos

Given metadata consisting of a title and a short textual description of an event, such as the one shown in Figure 3, our goal is to learn a visual classifier for it. Our approach begins with collecting a set of videos from the web for training. This set is obtained by a YouTube search with queries generated automatically from the metadata (Section III-A). We prune this collection to account for noisy data, which are common in any web-based data retrieval (Section III-B). Then, we represent each video with state-of-the-art convolutional neural network (CNN) features [36] and dense trajectories [42], and learn event classifiers (Section III-C).

[1] The '0' denotes the number of training examples provided.

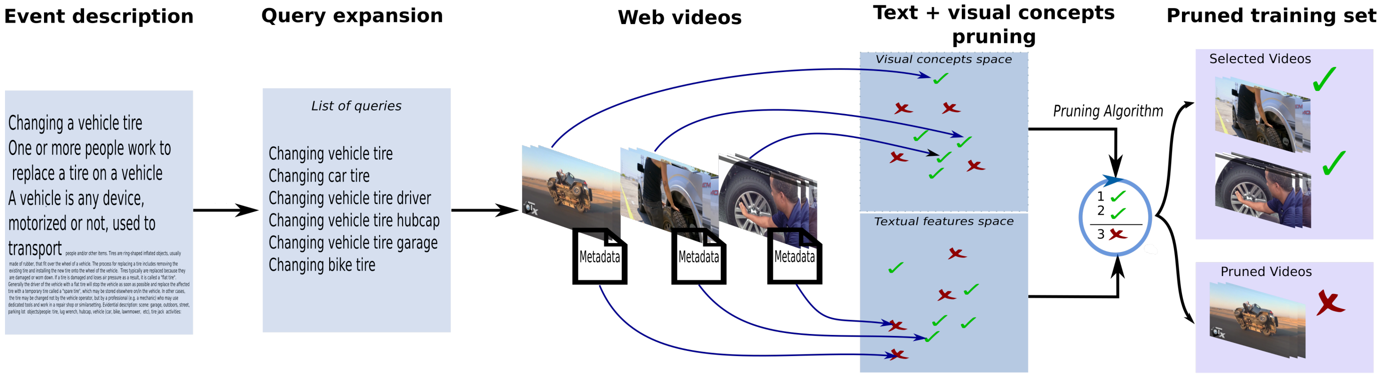

Event description
Query expansion
Web videos
Text + visual concepts pruning
Pruned training set
Changing a vehicle tire
One or more people work to replace a tire on a vehicle
A vehicle is any device, motorized or not, used to transport people and/or other items. Tires are ring-shaped inflated objects, usually made of rubber, that fit over the wheel of a vehicle. The process for replacing a tire includes removing the existing tire and installing the new tire onto the wheel of the vehicle. Tires typically are replaced because they are damaged or worn down. If a tire is damaged and loses air pressure as a result, it is called a "flat tire". Generally the driver of the vehicle with a flat tire will stop the vehicle as soon as possible and replace the affected tire with a temporary tire called a "spare tire", which may be stored elsewhere on/in the vehicle. In other cases, the tire may be changed not by the vehicle operator, but by a professional (e.g. a mechanic) who may use dedicated tools and work in a repair shop or similarsetting. Evidential description: scene: garage, outdoors, street, parking lot objects/people: tire, lug wrench, hubcap, vehicle (car, bike, lawnmower, etc), tire jack activities:
List of queries
Changing vehicle tire
Changing car tire
Changing vehicle tire driver
Changing vehicle tire hubcap
Changing vehicle tire garage
Changing bike tire
Metadata
Metadata
Metadata
Visual concepts space
Textual features space
Pruning Algorithm
1 ✓
2 ✓
3 ✗
Selected Videos
Pruned Videos

Fig. 3. Overview: Given the description of an event ("Event description"), relevant queries are automatically generated ("Query generation") to collect an initial training set ("Web videos"). Text metadata and visual concepts extracted from these videos are used to select the relevant ones automatically ("Text + visual concepts pruning"), and build a training set for event classification ("Pruned training set").

A. Textual query generation

A straightforward way to generate queries is to use the title of the event, e.g., *birthday party*, *changing vehicle tire*. While this is a good starting point to retrieve relevant data from the web, it falls short on using the rich content in the description of the event effectively. For example, events such as *birthday party* involve various objects like cake, candle, and can be organized as a barbecue, tea party, ball, etc. In the case of the event *changing vehicle tire*, changing a car tire is very different to changing bicycle tire. Such content is typically available in the event description. In order to better exploit this rich text information, we design a query expansion strategy.

We start with the event description available in TrecVid data, which contains a definition—a short phrase of 5 to 10 words, and a short paragraph describing the event and the context in which the event occurs (see the *Changing vehicle tire* example in Figure 3). A similar description can also be provided by a user when defining the list of events. The description presents scenes and objects potentially involved, activities that are likely to occur, and in some cases a textual description of audio in the event. We create queries specific to each event from all of this text.

The event title forms the reference query in our approach. It is used to automatically produce additional queries as described in the following. We first tag each word in the event description (i.e., title, and all the associated text) with the Stanford part of speech tagger (Stanford POSTagger) [41]. This assigns every word one of the 31 standard tags, such as noun-singular, adjective, adverb. We then compute a similarity between words in the event title and those in the remaining (longer) event description. For example, in the case of the event *changing vehicle tire*, we compute the similarity s between every word in the title tw (i.e., changing, vehicle, tire) and all the words w of the same tag type in the text description, individually. In this example, "changing" is compared to other verbs in the event description. The similarity s is given by:

$$s(tw, w) = \frac{1}{2}(\mathbf{x}_w \cdot \mathbf{x}_{tw} + \frac{1}{N}\sum_{\substack{i=1 \\ tw_i \neq tw}}^{N} \mathbf{x}_w \cdot \mathbf{x}_{tw_i})), \quad (1)$$

where $\mathbf{x}_w$ and $\mathbf{x}_{tw}$ are feature representations of words w and tw representations, N is the total number of words in the event title, with tw_i denoting the i th word from it. The feature representation is computed with Word2Vec [28], which is an embedding of a word into a vector space. More details of this feature computation are presented in Section IV-B.

Dot product between two feature vectors is a measure of the similarity between the two corresponding words, as used in [28]. The first term in (1) measures the similarity between a word in the event title, tw, and one of the words from the event description, w. The second term measures the similarity between w and the remaining words in the event title, denoted by tw_i. For the *changing vehicle tire* class, this would mean a high similarity to "tire" as well as "changing", when finding words similar to "vehicle". We then retain all the words of the same type with high similarity. To sum up, we use verbs, nouns and adjectives as reference words in the event title, and generate a set of event-related words for each tag-type. In the case of *changing vehicle tire*, the set of relevant verbs associated with "changing" contains "replacing", and the set of nouns related to "vehicle" contains "car", "bike". These sets of words are then used for query expansion based on the following three strategies determined by the structure of the event title.

(i) **Event title contains verb and object.** If a word from the relevant set is a hyponym of the reference word [29], a new query is created by substituting the reference word with the related word. Otherwise, the word is added to the original query. For example, since words like "car", "bike" are hyponyms of vehicle, the query "changing vehicle tire" becomes "changing bike tire" and "changing car tire". With "hubcap", the new query is "changing vehicle tire + hubcap". The intuition behind this expansion strategy is to adapt queries to intra-class variation.

(ii) **Event title without verb.** We first generate additional queries with the strategy described above, for the two other tag types, i.e., noun and adjective. Since action events are better described with verbs, we propose an additional strategy to compensate for the lack of verbs in the event title. We

TABLE I
QUERY EXPANSION RESULTS FOR SIX OF THE TRECVID13 CATEGORIES.

Changing vehicle tire	Parkour	Cleaning an appliance
Changing vehicle tire Changing car tire Changing vehicle tire driver Changing vehicle tire wheel Changing vehicle tire lawnmower Changing vehicle tire hubcap Changing vehicle tire garage Changing vehicle tire person Changing bike tire replace vehicle tire	Parkour Parkour parkour Parkour skateboarding Parkour snowboarding Parkour gymnastic Parkour acrobatic Parkour outdoor Parkour maneuver	Cleaning an appliance Cleaning an appliance household Cleaning an appliance cooker wash an appliance Cleaning an dishwasher Cleaning an dryer Cleaning an toaster Cleaning an refrigerator

Dog show	Rock climbing	Town hall meeting
Dog show Dog show show Dog show kennel Dog show leash Dog show obedience Dog show breed Dog show handler Dog show Frisbee chihuahua show sheepdog show	Rock climbing Rock climbing climb Rock climbing climber Rock climb Rock climbing jump Rock climbing wall	Town hall meeting Town hall meeting village Town hall meeting auditorium Town hall meeting community Town hall meeting discuss Town hall meeting event Town hall meeting vote Town hall meeting discussion Town hall meeting attend

use the similarity measure (1) to find verbs related to words of other tag-types in the title. Each of these related verbs is individually added to the title to produce new queries. Consider the *birthday party* class, which has no verb in the event name, as an example. An automatically extracted verb related to this event, “celebrate”, is added to the event title to generate a new query “birthday party celebrate”.

(iii) **Event title with a single word.** We handle events such as *parade*, *parkour*, which contain a single word in the event title, separately. In order to avoid drifting from the original semantic meaning, which is likely to occur when we replace the only title word with those related to it, we propose adding each one to the title to generate new queries, instead of replacing. In the case of the reference query “parkour”, it is related semantically to “gymnastics”. However, replacing it with “gymnastics” as the new query results in a large number of generic videos of gymnastics, and not all of them belong to the *parkour* class. We avoid this with new queries targeted to events, i.e., “parkour gymnastics” in this case.

At the end of the query expansion step, we have a rich set of event-related queries (see examples in Table I). These automatically generated queries allow for the creation of a new event dataset with web resources. In this work, we download videos and their corresponding metadata from YouTube. Implementation details of our query generation method are presented in Section IV-B.

B. Pruning with text and visual classification

Given the variety of data available online, we observed that more than half of the videos are irrelevant to an event. For example, videos in which a person is talking about *changing a car tire* without actually doing it, or videos displaying a *parkour* in the video game Minecraft, or videos of red carpet arrivals for a celebrity *birthday party*, are available through YouTube, but are not relevant to learn the action event. To prune such irrelevant videos, we use text and visual concept features jointly. We begin by describing the representation of text and visual data with tf-idf and visual concept features respectively, and then present our pruning algorithm.

Tf-idf features. We represent text data with tf-idf (term frequency-inverse document frequency) features [12], which have achieved excellent performance for text classification [17]. These features capture the importance of a word to a document in the corpus. In our case, the corpus is the set of YouTube text metadata of all the downloaded videos, the text associated with each video is the document, and the set of words occurring in the corpus is the dictionary. Two weight vectors are computed for each document with the metadata vocabulary: tf and idf. Term frequency (tf) measures the number of occurrences of a word from the dictionary in a document. Inverse document frequency (idf) is the proportion of documents in which a word appears in the corpus. In other words, it measures how much information a word provides, in terms of it being common or rare in the corpus. We compute tf and idf for each word in the dictionary and compose them into two vectors. The tf-idf feature vector is the element-wise product of these vectors. This combination of tf and idf vectors diminishes the importance of words that occur frequently in the corpus, as they are not relevant for distinguishing documents, and on the other hand, increases the influence of rare words.

Visual concept features. We compute visual features from video data to complement text features extracted from metadata. This helps leverage the visual similarities among videos depicting the same event. For example, in videos of the event *changing vehicle tire*, a tire is visible in at least a part of the video. To this end, we use state-of-the-art convolutional neural networks [20], [36]. In particular, we use the VGG-16 network [36] trained on ImageNet with 1000 classes. The last layer of this network (fc8) is a soft-max score indicating the presence of a class. We refer to these classes as visual concepts [1] as they encode semantic content based on the appearance in input images. A visual concept can be an object, a place, an animal, or a texture. We compute the visual concept scores for each image independently, and aggregate the number of activations, i.e., the number of non-zero probabilities, of each concept over the entire video.

Pruning algorithm. Given the text and visual representations of videos collected with our queries, we present a two-step approach to prune them. In the first step, we perform pruning with text data to select an initial set of relevant examples. As demonstrated in our experimental analysis, this step removes some of irrelevant videos. To leverage the complementary cues in visual data, we perform a second pruning step with visual concept features.

The first step uses the given textual description of events, as it is the primary source of information about an event. For each event, all the videos downloaded are ranked with tf-idf features, by comparing them to the tf-idf representation of the event description. Specifically, we compute the dot product of event description and metadata tf-idf features. The top-ranked examples with this are videos whose metadata is most similar to the event description, in terms of word statistics.

These videos are considered as representative examples for the event. We take the top-ranked videos chosen with a threshold on the dot product matching score, determined empirically, to evaluate this text-only pruning step (see "txt prun." in Table II). Note that this threshold is independent of the event type, and its impact on the performance is analyzed in Section IV.

To also prune with visual data, we compute the mean visual concept feature vector of the top-20 videos ranked with text. In other words, we take the most reliable videos in terms of their textual description, and extract a representation of the occurrence of visual concepts in the event. We then measure the relevance of any video to an event as the dot product of this mean vector and the video's visual concept feature. We re-rank all the video examples according to this similarity measure, and re-compute the mean vector with the new top-20 videos. This step is repeated several times, until the set of top-20 videos does not change, typically less than 50 iterations. This visual data pruning allows us to retrieve videos similar in visual content even if they are lacking in metadata, for example, when it is not available for a video. On the other hand, it also prunes videos with good text description, but no relevant video content, e.g., a person talking about how to change a car tire, without actually demonstrating how it is done in the video clip.

C. Video description and classifier

At the end of the pruning stage, we have an automatically refined set of videos to learn the event classifiers. We use a combination of CNN [36] and dense trajectory features to represent these videos. The CNN features are computed per frame and mean-pooled over time. For the dense trajectory features, we compute HOG, HOF, MBH descriptors along each trajectory, and aggregate them into a Fisher vector, as described in [42]. These two features are complementary—while CNN features describe appearance at the frame level, dense trajectories extract motion information in the video. For example, in a *parade* video, dense trajectories capture the movement of people walking in one direction, while CNN extracts visual attributes like people, flags, costumes, etc. Using these two features together, by concatenating them into a single vector, rather than separately leads to a significant gain in performance. We demonstrate this empirically in Table III (see Section IV-C for details). Given the set of automatically selected positive training video samples, we learn a one-vs-rest linear SVM for each class, with all videos from the other classes as negative samples.

IV. Experiments

A. Datasets

We used videos from the TrecVid multimedia event detection task to evaluate our approach. In particular, we used test videos from the 2011 challenge to analyze the variants of our method: different video feature representations, using only the reference queries, and the complete approach with query expansion. We take the best variant from these (i.e., full method with query expansion) and evaluate it on the 2013 challenge videos. Note that none of the TrecVid training videos were used in our experiments, except for the analysis "Adding TrecVid11 training data to the pruned set" in Section IV-C. We follow the standard TrecVid evaluation protocol and report mean average precision (mAP).

The TrecVid 2011 (TrecVid11) dataset contains videos of 10 events: *birthday party*, *changing vehicle tire*, *flash mob gathering*, *unstuck vehicle*, *grooming an animal*, *making a sandwich*, *parade*, *parkour*, *repairing an appliance*, *sewing project*, along with videos unrelated to any of these classes, i.e., *background* category. These classes are referred with labels E006 for *birthday party*, E007 for *changing vehicle tire*, ..., and E015 for *sewing project*. The test set contains 1244 videos of the 10 events, and 30,576 for the background class.

TrecVid 2013 (TrecVid13) contains 10 additional events: *attempting bike trick*, *cleaning an appliance*, *dog show*, *giving directions to location*, *marriage proposal*, *renovating home*, *rock climbing*, *town hall meeting*, *winning race without vehicle*, *working on metal crafts*. These classes are assigned labels E021 (for *attempting bike trick*) through E030 (for *working on metal crafts*). The test set has a total of 24,957 videos, among which 23,468 belong to the background class. We follow the EK0 challenge protocol for this dataset, where no training videos are available to learn the event classifiers.

B. Implementation details

Query generation and expansion. During the query generation process (Section III-A), a word from the event description is selected to create a new query if its similarity with a word from the event title, according to (1), is greater than 0.35. Table I shows a few examples of queries generated. For computational reasons, we limit the maximum number of queries to 10, including the reference query. If the query generation step produces more than 10 queries, we pick the top ones most similar to the event title, according to the measure (1). We download 150 videos for the reference query, and 50 each for other queries from YouTube, along with their corresponding metadata. This forms our initial training set.

For TV11 events, 3626 videos were downloaded with our query expansion. The pruning algorithm selects 2172 videos from this set. Note that the pruning step only changes the number of positive samples for each class, and all the videos downloaded from the other classes, pruned or not, form the negative exemplar set.

Text data. We use metadata fields "tags" and "description" available with YouTube videos to form the text component of our training data. We build a dictionary of words from all these text metadata files, by applying standard text processing techniques, such as stemming [26] and removing stop words. This results in a 8652-word dictionary for TrecVid 2011, and one with 8000 words for TrecVid 2013. We compute a 8652-dimensional (8000-dim for 2013) tf-idf representation for each text metadata document (as described in Section III-B). Idf is computed over all categories jointly. We also apply sublinear tf scaling [40], which replaces tf by:

$$\text{tf} = \begin{cases} 1 + log(\text{tf}) & \text{if tf} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

TABLE II
INFLUENCE OF OUR PRUNING APPROACHES ON THE TRECVID11 TEST SET. WE SHOW THE RESULT USING THE INITIAL TRAINING SET ("NO PRUN."), AND THE TRAINING SET PRUNED WITH TEXT ("TXT PRUN."), TEXT AND VISUAL FEATURES ("TXT+VIS. PRUN."). THE VARIANT "REF. QUERY" IS ONE WHERE ONLY VIDEOS OBTAINED WITH THE REFERENCE QUERY ARE USED TO TRAIN THE EVENT CLASSIFIERS.

Category	Ref. query	Query expansion		
		no prun.	txt prun.	txt+vis. prun.
E006	8.93	16.65	17.50	19.87
E007	48.60	54.67	51.45	55.58
E008	21.62	29.27	38.29	42.36
E009	42.00	35.13	33.23	40.89
E010	13.35	11.21	14.19	15.08
E011	10.24	12.29	10.55	18.04
E012	15.03	38.13	36.81	45.37
E013	22.93	28.77	37.83	30.67
E014	23.09	30.87	32.00	36.57
E015	21.65	22.87	24.12	22.91
mean	22.74	27.99	29.60	32.73

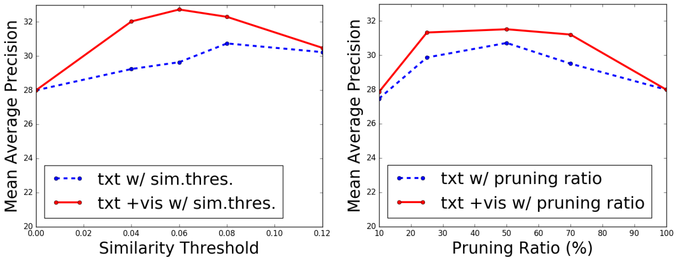
TABLE III
EVALUATING DENSE TRAJECTORY ("DENSE TRAJ.") AND CNN ("VGG-16") FEATURES, AND THEIR COMBINATION ("COMBINED") AS A VIDEO REPRESENTATION ON THE 10 EVENT CLASSES (E00X) FROM THE TRECVID 2011 TEST SET. SEE TEXT IN SECTION IV-C FOR DETAILS.

Category	dense traj.	VGG-16	Combined
E006	14.51	19.07	19.87
E007	31.80	49.62	55.58
E008	39.29	27.09	42.36
E009	23.25	37.48	40.89
E010	6.22	8.96	15.08
E011	7.74	13.18	18.04
E012	37.38	35.51	45.37
E013	24.87	21.14	30.67
E014	22.04	26.64	36.57
E015	15.78	13.70	22.91
mean	22.29	25.23	32.73

This feature is then L2-normalized after multiplying tf and idf. We use the same dictionary to compute the tf-idf representation of each TrecVid event description.

Video data. One of the issues with downloading videos from YouTube for events like *birthday party* is the presence of animated slideshows. Such videos are a collection of non-contiguous photos or title slides (e.g., "Happy Birthday!") and lack any movement depicting an action. To remove these slideshows from our dataset, we measure the similarity between two consecutive frames with the L2-norm of the difference of their appearance features. If this norm is less than 0.15, the frames are considered as similar. If two consecutive frames i and $i+1$ are similar, and frames $i+1$ and $i+2$ are also similar, then the three frames i, $i+1$, $i+2$ are grouped into a same set of similar frames. With this method, if a video contains N slides, its frames are split into N sets of similar frames.

Visual concept and video features. We use the VGG-16 network to compute these features. It is composed of 13 convolution (with ReLU), 5 max pooling and 3 fully-connected layers, and an additional soft-max layer. All the convolutional filters are of size 3×3. The network is trained with ImageNet on 1000 categories to extract our visual concept as well as video features as follows. We compute VGG-16 responses for one in every ten frames independently, normalized them with their L2-norm, and pool them temporally to get a video representation. Specifically, we use the 1000-dimensional output of the fc8 layer for the visual concept feature. These correspond to ImageNet categories, which are objects (*lifeboat*, *convertible*, *bassoon*), places (*restaurant*, *home theater*), animals (*gazelle*, *sea lion*), or textures (*velvet*). For extracting video features to learn the classifiers (as described in Section III-C), we use richer features from the fc6 layer.



(a) (b)

Fig. 4. Impact of (a) the similarity threshold and (b) the pruning ratio are shown as mean average precision on the TrecVid11 test set.

In the "txt+vis. prun." method, the textual ranking from "txt. prun." method is used as initialization. For a frame i, we compute the VGG16-fc8 feature, $fc8_i$. For computational reasons, this feature is extracted every ten frames. We then build a matrix $M_{fc8} \in \mathbf{R}^{1000 \times N}$, where the i-th column is the $fc8_i$ feature vector, and N is the number of frames. We then aggregate these frame-level features temporally to obtain the visual representation, with a L0-norm pooling operation in the temporal dimension. L0-norm pooling measures the number of times a concept is activated in the video, and performs better than L1-norm pooling as it captures finer details, whereas L1-norm is more sensitive to large activations. For example, for the *changing vehicle tire* event, the "tire" concept gives large values all over the video, reducing the importance of other concepts if L1-pooling is used. This produces a 1000-dimensional vector which describes the video in the visual concept space. This feature is then L2-normalized.

We reduce the dimension of the HOG, HOF, and MBH components of dense trajectory features by a factor of 2 with principal component analysis. We then perform power and L2 normalization, and use 256 Gaussian components for the mixture model.

Classifier. The SVM for video classification is implemented with LIBSVM [2]. Regularization and class imbalance parameters are set with 10-fold cross-validation. During cross-validation, the training set is split randomly into a validation set, and a smaller training set containing 75% of the training data, whilst maintaining the original ratio of positive and negative samples. Mean average precision on the validation set is computed for each choice of parameters, and the one

TABLE IV
THE 10 MOST RELEVANT VISUAL CONCEPTS FOR FOUR EVENTS. THESE CORRESPOND TO THE CONCEPTS WITH THE TOP-10 VALUES IN THE MEAN VISUAL CONCEPT VECTOR, COMPUTED AT THE END OF THE PRUNING ALGORITHM. SEE SECTION III-B FOR DETAILS.

Changing vehicle tire	*Unstuck vehicle*
1 - car wheel	1 - jeep, landrover
2 - minivan	2 - pickup, pickup truck
3 - car mirror	3 - tow truck, tow car, wrecker
4 - limousine, limo	4 - minivan
5 - crash helmet	5 - snowplow, snowplough
6 - disk brake, disc brake	6 - minibus
7 - vacuum, vacuum cleaner	7 - ambulance
8 - minibus	8 - golfcart, golf cart
9 - seat belt, seatbelt	9 - beach wagon, station wagon, wagon
10 - motor scooter, scooter	10 - car wheel

Repairing an appliance	*Grooming an animal*
1 - switch, electric switch, electrical switch	1 - hair spray
2 - washer, automatic washer, washing machine	2 - hand blower, hair dryer, hair drier
3 - refrigerator, icebox	3 - fur coat
4 - safe	4 - wig
5 - microwave, microwave oven	5 - Afghan hound, Afghan
6 - cash machine, cash dispenser	6 - swab, swob, mop
7 - dishwasher, dish washer,	7 - iron, smoothing iron
8 - loudspeaker, speaker, speaker unit	8 - cash machine, cash dispenser,
9 - iPod	9 - dishwasher, dish washer
10 - soap dispenser	10 - vacuum, vacuum cleaner

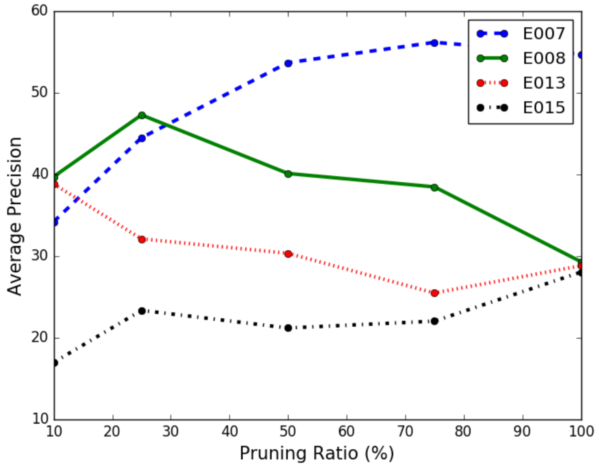


Fig. 5. Average precision for four classes with the “txt+vis. prun.” method using pruning ratio on the TrecVid11 test set.

with the best performance is chosen.

C. Results on TrecVid 2011

Our overall method, where classifiers are learned with CNN and dense trajectory features computed on the (two-step) pruned video set, achieves 32.73 mAP, as shown in Table II. We use TrecVid 2011 as a test-bed to analyze several variants of our approach.

Importance of query expansion. Table II shows a comparison of variants based on the queries used to constitute the training data. We report results for query expansion (shown as in “Query expansion” in the table), where we download additional videos with our textual query expansion. We observe that query expansion significantly improves over the baseline reference query result (shown as “Ref. query” in the table), by over 5% in mAP score on average. For E012 (*parade*), query expansion improves AP from 15.03 to 38.13, due to the creation of accurate additional queries (e.g., “parade procession”, “parade march”, “parade commemoration”). Query expansion can affect the performance negatively in a few cases. For E009 (*unstuck vehicle*), we see a reduction from 42.00 to 35.13, as expansion adds noisy videos due to the addition of generic verbs, e.g., “have”, “do”, to the initial query. This behavior is more of an exception than a rule. Pruning methods do compensate for most of this loss, with the final result of 40.89 for this event.

Importance of pruning algorithm. The method “txt prun.” in Table II is the variant where only text features are used to prune the initial set of videos created with our query expansion (see Section III-B for details). This further improves the mAP over the “no prun.” variant by 1.61%. The overall method, “txt+vis. prun.”, which uses text and visual features for pruning gives an additional gain of 3% in mAP score over text-only pruning. We performed an additional experiment by manually annotating videos for two classes: E006, E008. We built a positive set for the two classes by watching all the videos, and annotating relevant ones as positive samples. The results for this ground truth annotation are 22.13, 46.23 respectively. Our result of 19.87 and 42.36 for these classes, using no annotated data, is close to this upper bound, further highlighting its effectiveness.

We also analyze the 10 most relevant visual concepts for the four sample events in Table IV. They correspond to the concepts with the most important activations in the mean visual concept vector, which is computed over the top-20 ranked videos at the end of our pruning algorithm. We observe that concepts are semantically related to the event. For example, “car wheel” is the most important concept for the event *Changing vehicle tire*, and the top-10 relevant concepts for the event *repairing an appliance* include appliances such as washer, refrigerator. While the events *changing vehicle tire* and *unstuck vehicle* can be visually similar (see examples in Figure 2), the relevant concepts help us distinguish them, e.g., “snowplow” is relevant only for *unstuck vehicle*.

Combining appearance and motion features. We analyze the performance of three visual feature representations: dense trajectories, VGG-16 fc6 responses, and a combination of the two features, in Table III. The variant with VGG-16 features performs better than dense trajectories on average. Dense trajectories, however, show a better performance for four events, *flash mob gathering*, *parade*, *parkour*, *sewing project*, where motion plays an important role in representing them. Combining the two features, by stacking them into a single vector, outperforms using either of the two features individually by a large margin—an improvement of over 7% on average. The classifiers learn the relative importance of these two features automatically from the training data. Significant improvement is observed for all the TrecVid11 events due to the two representations being complementary. For example, for a *changing vehicle tire* video, dense trajectories capture the movement of a person manipulating the car jack, while VGG-16 extracts visual attributes like tire,

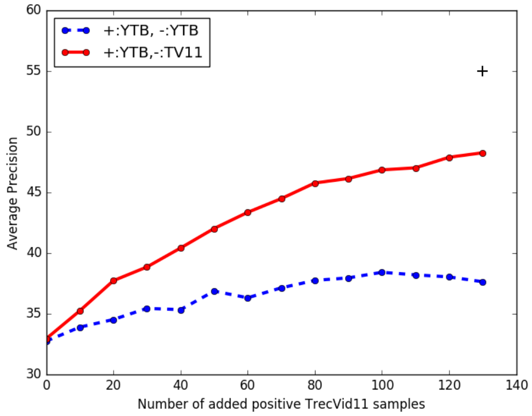


Fig. 6. Impact of adding TrecVid11 positive and negative samples. "+:YTB, -:YTB" is the pruned YouTube dataset to which TrecVid samples are added. "+:YTB, -:TV11" is the variant with pruned YouTube samples as positive and background TrecVid11 videos as negative samples, with progressive addition of TrecVid11 positives. The average precision of a method trained on the entire TrecVid11 training set is shown with a black cross.

garage, car. For a *parkour* video, dense trajectories capture the movement of a person performing activities, while VGG-16 extracts visual attributes like body shape, streets, parks. Thus, we use the combined feature vector in our full method.

Impact of the similarity threshold. The similarity threshold determines the number of videos pruned with tf-idf (see the details of pruning in Section III-B). We analyze its impact by varying the threshold on TrecVid11 in Figure 4(a). We observe best performance on this dataset for a threshold value of 0.06. We also followed an alternate strategy of using a pruning ratio, i.e., keeping $r\%$ of the downloaded videos, to determine the refined set used for training. This is shown in Figure 4(b). Note that pruning ratio 100 and the similarity threshold 0 are equivalent to the "no prun." method. On average, using a similarity threshold performs better than using a pruning ratio. In the latter method, the number of positive samples depends on the total number of downloaded videos, but many videos can be relevant for some events (see E007, E015 in Figure 5), and irrelevant for other events (see E008, E013 in Figure 5). Using an absolute threshold to determine the pruned set, as in the similarity threshold strategy, maintains a uniform quality for this pruning approach over all the events. We observe that "txt+vis. prun." outperforms "txt. prun." in both (a) and (b) in the figure.

Adding TrecVid11 training data to the pruned set. We study the impact of introducing TrecVid11 training dataset into our (pruned) training set through two experiments, presented in Figure 6. First, we progressively add positive samples from TrecVid11 training dataset to our training set composed of pruned videos. This results in an improvement from 32.73 (the overall result in Table II) to 38.39 shown in the dashed-line (blue) curve in the figure, for 100 positive TrecVid samples added per event. This is partly due to bias in the TrecVid11 dataset, wherein videos in the training and test sets are very similar. For example, videos in the training and test set[2] for the *making a sandwich* event show the same person in the same kitchen, and from the same viewpoint.

TABLE V
PER-CATEGORY RESULTS FOR ZERO-SHOT LEARNING ON TRECVID MED2013 EK0 TEST SET.

Category	[37]	[7]	Query expansion		
			no prun.	txt prun.	txt+vis. prun.
E006	14.48	15.148	18.12	20.70	25.86
E007	41.37	39.60	60.78	57.94	66.39
E008	45.56	19.30	42.44	47.66	51.22
E009	53.71	36.80	47.66	49.15	59.12
E010	6.38	8.60	13.99	13.27	20.41
E011	11.56	15.10	16.15	19.79	16.91
E012	17.76	32.20	54.39	53.30	55.16
E013	7.86	12.90	45.84	57.03	58.33
E014	15.18	16.10	35.88	40.68	47.68
E015	3.23	29.20	24.19	31.70	31.79
E021	6.76	6.60	9.81	6.94	6.94
E022	3.11	2.10	16.07	14.76	19.91
E023	0.91	40.50	38.46	37.32	38.29
E024	0.55	1.60	10.28	10.06	13.15
E025	0.21	1.30	8.57	16.54	10.96
E026	3.63	3.90	5.02	5.18	2.86
E027	1.44	13.20	15.19	15.84	13.88
E028	0.95	10.50	22.79	29.86	35.96
E029	0.10	13.70	0.20	0.61	0.40
E030	0.82	2.90	0.45	1.25	0.60
mean	11.81	16.10	24.29	26.44	28.79

In the second experiment, we study the impact of using 9600 videos corresponding to "background" from the TrecVid11 training set. We follow the standard TrecVid protocol of using these videos as negative examples. We add positive samples from the TrecVid11 training set to a dataset composed of our positive pruned YouTube samples and TrecVid11 background events as negative exemplars. When no positive TrecVid11 samples are added, we obtain an mAP of 32.92 (the lowest point on the red solid-line curve in Figure 6), which is very similar to the result with our pruned dataset (32.73). Adding positives from TrecVid11 in this case, we observe that the improvement is more significant than in the first case (blue dashed-line in the figure). This is potentially due to domain adaptation issues, where TrecVid11 positive samples are visually more similar to the negative ones than our YouTube positive exemplars. Videos in the YouTube set are longer, have a higher resolution and more professional content (e.g., tutorial for changing vehicle tire or making a sandwich), than the TrecVid11 examples. For comparison, we also show the performance (55.01 mAP) when learning the classifiers with manually-annotated TrecVid11 training set (black cross in the plot). The difference between this and our final result of adding all TrecVid11 positives to the YouTube set (48.24 on the red solid-line curve in the figure) is also due to the domain adaptation problem between TrecVid and YouTube videos.

[2] Videos named HVC494077, HVC516890, HVC606263, HVC788253 from TrecVid11 training set and videos HVC220966, HVC358422, HVC122406, HVC648049, HVC218518 from the test set.

TABLE VI
COMPARISON TO THE STATE OF THE ART FOR ZERO-SHOT LEARNING ON TRECVID MED 2013 EK0 TEST SET. VIDEO REPRESENTATIONS USED BY EACH METHOD (DENOTED BY "✓"), APPEARANCE FEATURES ("APPEARANCE"), MOTION FEATURES ("MOTION"), AND SPEECH OR TEXT FEATURES ("SPEECH/TEXT"), ARE ALSO SHOWN.

Methods	[3]	[14]	[44]	[10]	[37]	[16]	[15]	[7]	[46]	[9]	Ours: Query expansion		
											no prun.	txt prun.	txt+vis. prun.
appearance	✓	✓	✓	–	✓	✓	✓	✓	✓	✓	✓		
motion	–	–	✓	✓	–	✓	✓	✓	✓	✓	✓		
speech/text	–	✓	–	–	–	✓	✓	–	–	–	–		
mAP	2.30	2.50	6.12	6.39	11.81	20.80	22.12	16.1	8.86	16.7	24.29	26.44	**28.79**

D. Results on TrecVid 2013

Having used TrecVid11 as a test-bed to evaluate all the variants of our method, we choose the best-performing method on it (i.e., the method using query expansion, combination of appearance and motion features, and a similarity threshold of 0.06) as our full method. Table V shows the results of our full method with text and visual feature pruning ("txt+vis. prun." in the table). The method "txt prun." shows an improvement on 2.20% in mAP score over "no prun.". The full method "txt+vis. prun." gives an additional gain of 2.3% in mAP, over text-only pruning. We also report results from [37], which is the only recent method providing per-class results on TrecVid13. We added per-class results provided by the authors of a very recent paper [7]. Our method outperforms [7], [37] by a large margin. For example, we obtain an average precision of 47.68 (16.10 for [7], 15.18 for [37]) for the *repairing an appliance* event. Figure 7 shows a few qualitative results of our approach on this dataset. All the displayed videos contain visual concepts related to the event: tire, and car jack for *changing vehicle tire*, cake and candles for *birthday party*, an appliance for *repairing an appliance*, a group of people wearing the same uniform for *parade*. It shows the importance of visual concepts for pruning.

E. Comparison to the state of the art

We compare with several approaches [3], [7], [10], [14]–[16], [37], [44] on the TrecVid 2013 EK0 challenge dataset. Here, we focused on methods which build a training set from internet data. As shown in Table VI, our results have a mean average precision of 28.79%, which is a significant gain of 6.5% over the state-of-the-art approach [15], which also curates a training set from the internet. We also compare with other recent methods: 20.80% [16], 16.7 [9], 16.1 [7], 11.89% [37], 8.86 [46], 6.39% [10], 6.12% [44], 2.5% [14], 2.3% [3]. We outperform all these methods significantly, due to the following key differences. We use motion information effectively, in contrast to [3], [7], [14], [37], [44], [46], relying on image data alone. Our query expansion method exploits critical elements of actions, e.g., related verbs, whereas [3], [7], [9], [10], [14]–[16], [37], [44], [46] are limited to queries which are not as rich.

V. SUMMARY

This paper introduces a novel approach for event classification, given only a textual description of the event. Our approach relies on textual query expansion specifically designed for actions, which allows us to collect significantly more videos than using only the event name. A pruning step creates a reliable training dataset of videos sharing semantic and visual content. We show state-of-the-art results on TrecVid MED 2011 and 2013 in the zero-shot learning framework.

ACKNOWLEDGMENTS

This work was supported in part by the ERC advanced grant ALLEGRO, the Indo-French project EVEREST (no. 5302-1) funded by CEFIPRA. We gratefully acknowledge the support of NVIDIA with the donation of GPUs used for this work.

REFERENCES

[1] A. Binder, W. Samek, K.-R. Müller, and M. Kawanabe. Machine learning for visual concept recognition and ranking for images. In *Towards the Internet of Services: The THESEUS Research Program*. 2014.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems and Technology*, 2011.

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

[4] L. Chen, L. Duan, and D. Xu. Event recognition in videos by learning from heterogeneous web sources. In *CVPR*, 2013.

[5] S. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014.

[6] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. *PAMI*, 2012.

[7] C. Gan, C. Sun, L. Duan, and B. Gong. Webly-supervised video recognition by mutually voting for relevant web images and web video frames. In *ECCV*, 2016.

[8] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, 2015.

[9] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei. You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In *CVPR*, 2016.

[10] A. Habibian, T. E. J. Mensink, and C. G. M. Snoek. Composite concept discovery for zero-shot video event detection. In *ICMR*, 2014.

[11] A. G. Hauptmann and M. A. Smith. Text, speech and vision for video segmentation: The informedia project. In *AAAI Fall Symposium*, 1995.

[12] D. Hiemstra. A probabilistic justification for using tf-idf term weighting in information retrieval. *International Journal on Digital Libraries*, 2000.

[13] N. Ikizler-Cinbis, R. Cinbis, and S. Sclaroff. Learning actions from the web. In *ICCV*, 2009.

[14] L. Jiang, T. Mitamura, S.-I. Yu, and A. G. Hauptmann. Zero-example event search using multimodal pseudo relevance feedback. In *ICMR*, 2014.

[15] L. Jiang, S.-I. Yu, D. Meng, T. Mitamura, and A. G. Hauptmann. Bridging the ultimate semantic gap: A semantic search engine for internet videos. In *ICMR*, 2015.

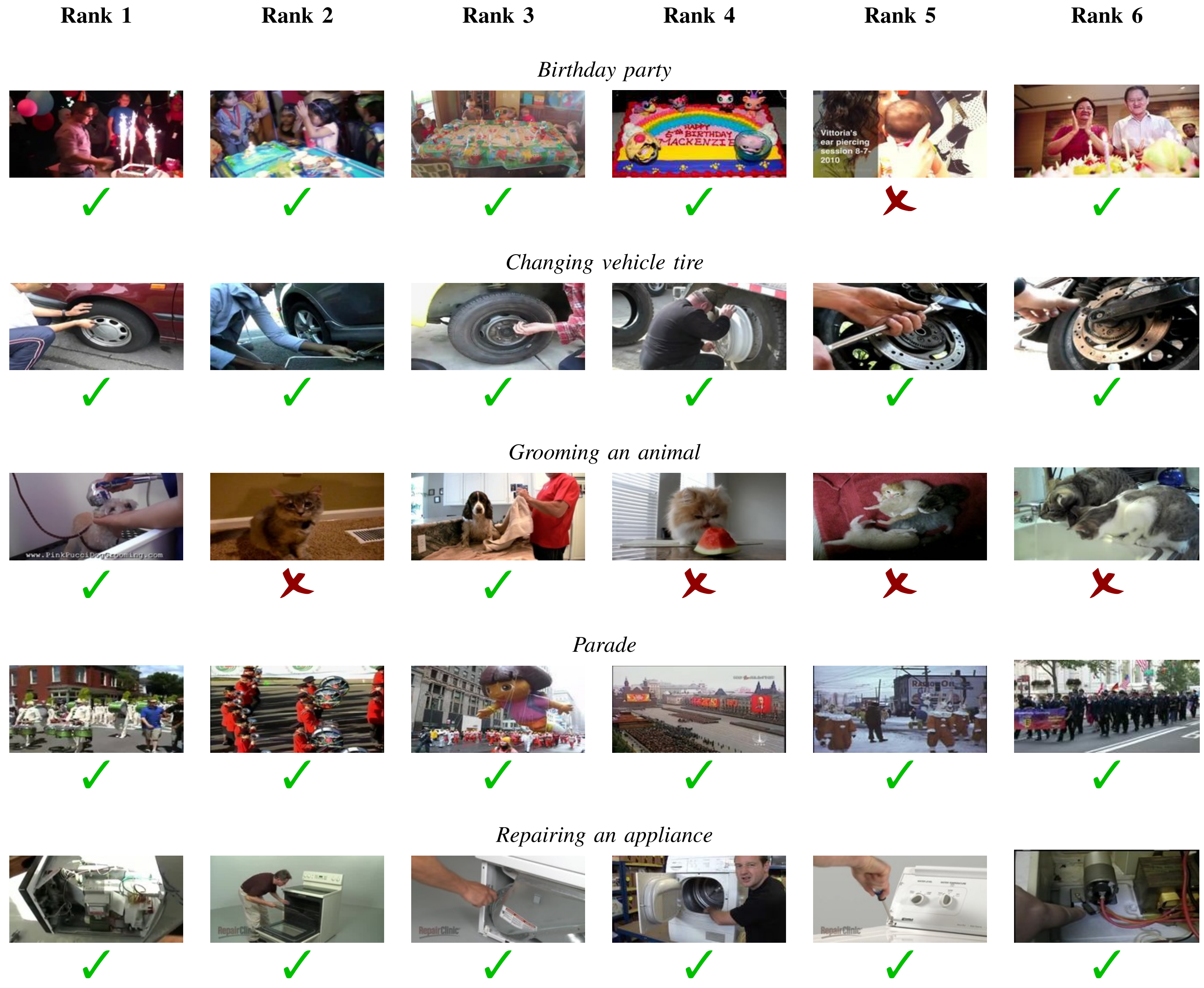


Fig. 7. The top-6 videos for the categories: *Birthday Party*, *Changing vehicle tire*, *Grooming an animal*, *Parade*, *Parkour*, *Repairing an appliance*, *Dog Show*, *Town Hall Meeting*, from the TrecVid 2013 test set, obtained with our method using automatically-curated training data.

[16] L. Jiang, S.-I. Yu, D. Meng, Y. Yang, T. Mitamura, and A. G. Hauptmann. Fast and accurate content-based semantic search in 100M internet videos. In *ACM Multimedia*, 2015.

[17] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *ECML*, 1998.

[18] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[19] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

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

[22] I. Laptev and P. Perez. Retrieving actions in movies. In *ICCV*, 2007.

[23] T. Leung, Y. Song, and J. Zhang. Handling label noise in video classification via multiple instance learning. In *ICCV*, 2011.

[24] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos “in the wild”. In *CVPR*, 2009.

[25] J. Liu, Q. Yu, O. Javed, S. Ali, A. Tamrakar, A. Divakaran, H. Cheng, and H. Sawhney. Video event recognition using concept attributes. In *WACV*, 2013.

[26] E. Loper and S. Bird. NLTK: The natural language toolkit. In *ACL Workshop*, 2002.

[27] T. Mensink, E. Gavves, and C. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[29] G. A. Miller. Wordnet: A lexical database for English. *Communications of the ACM*, 1995.

[30] P. X. Nguyen, G. Rogez, C. Fowlkes, and D. Ramanan. The open world of micro-videos. *arXiv:1603.09439*, 2016.

[31] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 2008.

[32] L. Niu, W. Li, and D. Xu. Visual recognition by learning from web data: A weakly supervised domain generalization approach. In *CVPR*, 2015.

[33] P. Over, G. Awad, J. Fiscus, B. Antonishek, M. Michel, A. Smeaton, W. Kraaij, and G. Quénot. TRECVID 2010 – An overview of the goals, tasks, data, evaluation mechanisms, and metrics. TRECVID, 2010.

[34] P. Over, J. Fiscus, G. Sanders, M. Michel, G. Awad, A. F. Smeaton, W. Kraaij, and G. Quénot. TRECVID 2013 – An overview of the goals, tasks, data, evaluation mechanisms and metrics. TRECVID, 2013.

[35] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014.

[37] B. Singh, X. Han, Z. Wu, V. I. Morariu, and L. S. Davis. Selecting relevant web trained concepts for automated event retrieval. In *ICCV*, 2015.

[38] M. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding techniques. In *CVPR*, 1997.

[39] Y. Song, M. Zhao, J. Yagnik, and X. Wu. Taxonomic classification for web-based videos. In *CVPR*, 2010.

[40] G. Töpper, M. Knuth, and H. Sack. DBpedia ontology enrichment for inconsistency detection. In *International Conf. Semantic Systems*, 2012.

[41] K. Toutanova and C. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *EMNLP*, 2000.

[42] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action Recognition by Dense Trajectories. In *CVPR*, 2011.

[43] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li. Youtubecat: Learning to categorize wild web videos. In *CVPR*, 2010.

[44] S. Wu, S. Bondugula, F. Luisier, X. Zhuang, and P. Natarajan. Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *CVPR*, 2014.

[45] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative CNN video representation for event detection. In *CVPR*, 2015.

[46] G. Ye, Y. Li, H. Xu, D. Liu, and S.-F. Chang. Eventnet: A large scale structured concept library for complex event detection in video. In *ACM Multimedia*, 2015.

Nicolas Chesneau is a PhD student at Inria, where he works in the Thoth team based in Grenoble. He received two MS degrees in computer science and applied mathematics, one from Telecom Bretagne, and one from École Normale Supérieure de Cachan.

Karteek Alahari is a tenured researcher at Inria, where he works in the Thoth team based in Grenoble. He received the BTech (with Honours) and MS degrees in computer science from IIIT Hyderabad, India, in 2004 and 2005 respectively, and the PhD degree from Oxford Brookes University, UK, in 2010. Prior to joining the team in Grenoble in 2013, he was a postdoctoral fellow in the WILLOW team at Inria Paris and École Normale Supérieure. He has been an associate member of the Visual Geometry Group at the University of Oxford, UK, since 2006, and also leads a research collaboration with Carnegie Mellon University, USA, since 2016. In February 2017, he was promoted to IEEE Senior Member.

Cordelia Schmid holds a M.S. degree in Computer Science from the University of Karlsruhe and a Doctorate, also in Computer Science, from the Institut National Polytechnique de Grenoble (INPG). Her doctoral thesis received the best thesis award from INPG in 1996. Dr. Schmid was a post-doctoral research assistant in the Robotics Research Group of Oxford University in 1996–1997. Since 1997 she has held a permanent research position at Inria Grenoble Rhone-Alpes, where she is a research director and directs an Inria team. Dr. Schmid has been an Associate Editor for IEEE PAMI (2001–2005) and for IJCV (2004–2012), editor-in-chief for IJCV (2013—), a program chair of IEEE CVPR 2005 and ECCV 2012 as well as a general chair of IEEE CVPR 2015 and ECCV 2020. In 2006, 2014 and 2016, she was awarded the Longuet-Higgins prize for fundamental contributions in computer vision that have withstood the test of time. She is a fellow of IEEE. She was awarded an ERC advanced grant in 2013, the Humbolt research award in 2015 and the Inria & French Academy of Science Grand Prix in 2016. She was elected to the German National Academy of Sciences, Leopoldina, in 2017.