Video2vec Embeddings Recognize Events when Examples are Scarce

Habibian, A.; Mensink, T.; Snoek, C.G.M.

DOI
10.1109/TPAMI.2016.2627563

Publication date
2017

Document Version
Final published version

Published in
IEEE Transactions on Pattern Analysis and Machine Intelligence

Video2vec Embeddings Recognize Events When Examples Are Scarce

Amirhossein Habibian, Thomas Mensink, and Cees G. M. Snoek, *Senior Member, IEEE*

Abstract—This paper aims for event recognition when video examples are scarce or even completely absent. The key in such a challenging setting is a semantic video representation. Rather than building the representation from individual attribute detectors and their annotations, we propose to learn the entire representation from freely available web videos and their descriptions using an embedding between video features and term vectors. In our proposed embedding, which we call *Video2vec*, the correlations between the words are utilized to learn a more effective representation by optimizing a joint objective balancing descriptiveness and predictability. We show how learning the Video2vec embedding using a multimodal predictability loss, including appearance, motion and audio features, results in a better predictable representation. We also propose an event specific variant of Video2vec to learn a more accurate representation for the words, which are indicative of the event, by introducing a term sensitive descriptiveness loss. Our experiments on three challenging collections of web videos from the NIST TRECVID Multimedia Event Detection and Columbia Consumer Videos datasets demonstrate: i) the advantages of Video2vec over representations using attributes or alternative embeddings, ii) the benefit of fusing video modalities by an embedding over common strategies, iii) the complementarity of term sensitive descriptiveness and multimodal predictability for event recognition. By its ability to improve predictability of present day audio-visual video features, while at the same time maximizing their semantic descriptiveness, Video2vec leads to state-of-the-art accuracy for both few- and zero-example recognition of events in video.

Index Terms—Event recognition, semantic video representation, representation learning

1 Introduction

This paper strives to recognize events such as *parking a vehicle*, *flash mob*, and *renovating a home* in web video content. A problem of increasing importance in a world that is swiftly adapting to video communication. The leading computer vision and multimedia retrieval solutions for this challenging problem, e.g., [1], [2], [3], [4], [5], all learn to recognize events with the aid of many labeled video examples. However, as events become more and more specific, we consider it unrealistic to assume that ample examples to learn from will be commonly available. In practical use only a handful of video examples, an event name and an event definition may be present, such as the ones in Fig. 1. We aim for event recognition when video examples are scarce or even completely absent.

Recognizing events from few or zero examples imposes constraints on the video representation. End-to-end deep learning of the video representation [6], [7], [8] demands too many video examples. In [8] for example, Karpathy et al. exploit more than 1 million YouTube videos and their sport category labels to learn the video representation. Alternatively, Xu et al. [1] show how a very deep convolutional neural network (CNN) [9] intended for image classification can be leveraged as representation for event recognition in video. They use responses from intermediate layers of the network to represent frames, which are aggregated over the video using VLAD encoding [10]. When combined with a linear SVM, excellent results on the leading NIST TRECVID event detection benchmarks [11] are reported for scenarios where many and few examples are available. The CNN video representation outperforms more traditional video encodings such as improved dense trajectories [2], [12] and representations combining appearance, motion and audio features [13], [14], [15]. However, both the learned and engineered representations are incapable, nor intended, to recognize events when examples are completely absent. We propose a video representation that can leverage any underlying feature, be it a CNN, improved dense trajectories and/or audio features, while being capable of few- *and* zero-example event recognition with state-of-the-art accuracy.

The key to few- and zero-example event recognition is to add meaning to the video representation. Inspired by the success in image classification [16], [17], [18], many rely on the predictions made by a set of attribute classifiers [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. In [23] Habibian et al. study the properties of 1,346 attribute classifiers trained from ImageNet [30] and TRECVID [11] for representing and recognizing events in web video. Rather than pre-specifying and manually labeling each individual attribute in advance, the attributes can be learned on top of imagery and (weak) labels harvested from the web [31], [32], [33], [34]. To assure visual predictability of the discovered attribute detectors a common tactic is to leverage part

- *A. Habibian was with the QUVA Lab, University of Amsterdam, Amsterdam 1012, WX, The Netherlands, and currently with Qualcomm Research, Amsterdam 1098, XH, The Netherlands. E-mail: habibian.a.h@gmail.com.*
- *T. Mensink is with the University of Amsterdam, Amsterdam 1012, WX, The Netherlands. E-mail: t.mensink@uva.nl.*
- *C.G.M. Snoek is with the QUVA Lab, Qualcomm Research, and University of Amsterdam, Amsterdam 1012, WX, The Netherlands. E-mail: cgmsnoek@uva.nl.*

Manuscript received 4 Nov. 2015; revised 21 May 2016; accepted 21 Sept. 2016. Date of publication 9 Nov. 2016; date of current version 12 Sept. 2017. Recommended for acceptance by R. Manmatha.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2016.2627563

Event Name: Winning a race without a vehicle

Definition: An individual (or more) succeeds in reaching a pre-determined destination before all other individuals, without vehicle assistance or assistance of a horse or other animal. Racing generally involves accomplishing a task in less time than other competitors. The only type of racing considered relevant for the purposes of this event is the type where the task is traveling to a destination, completed by a person(s) without assistance of a vehicle or animal. Different types of races involve different types of human ...

Fig. 1. Video exemplars (top) and the textual definition (bottom) of the event *winning a race without a vehicle* to illustrate one of the events studied in this paper. Following the NIST TRECVID evaluation guidelines [11], the textual definition is for zero-example recognition, and the ten provided video exemplars are for few-example event recognition.

of the harvested data for validation [35]. A drawback is that many attribute labels rarely occur. For these infrequent labels only a limited number of positive examples are available, which leads to a biased estimation of their classification reliability. Hence, many of the discovered attributes overfit to their small training set and do not generalize well for new videos. We also discover our semantic representation from the web, but rather than selecting individual, and often unreliable, classifiers per attribute label, we prefer to combine labels automatically into more predictable attributes. By doing so, more training examples are available and a more robust representation is obtained, without losing descriptive ability.

In this paper we present a semantic representation learning algorithm for videos. Instead of relying on pre-specified attribute labels, we learn the representation from freely available web videos and their descriptions. We propose an embedding between the video features and their textual descriptions, which is learned by utilizing the correlations between the words in the descriptions. We learn the embedding by minimizing a joint objective function balancing the descriptiveness and predictability of the learned video representation. Our embedding is able to leverage the multiple modalities which coexist in video to learn a more reliable semantic representation. Following the analogy from [36], we call our embedding *Video2vec*.

A preliminary version of this article appeared as [37]. This version adds i) zero-example event recognition ii) representation learning for multiple video modalities, iii) event specific representation learning, iv) an additional TRECVID video data set, revised experiments, and improved baselines, all using a recent video CNN feature [1], [38], and v) a new related work section.

2 Related Work

We focus on three directions of related work we deem most closely connected to ours.

2.1 Representations for Event Recognition

Until recently most event recognition methods exploit video representations based on densely extracted low-level visual features, such as HOG/HOF [39], or MBH [12], [40], often combined with audio features like MFCC features [4], [5], [41]. Currently, most methods extract frame-based deep convolutional neural network features [9], [38], [42], using the responses from intermediate layers of the CNN, which is pre-trained on ImageNet images [30], see, e.g., [1], [15], [29], [33], [43]. To obtain per video descriptors the local/frame-based descriptors are aggregated by their mean, by using the Fisher vector [44] or VLAD encoding [10]. Despite the fact that these low-level based representations obtain state-of-the-art recognition performance, they suffer from two drawbacks. First, because of their high dimensionality, training effective classifiers on these representations requires a sufficient number of training examples to prevent overfitting. Second, all these representations are incapable of providing a semantic interpretation of the video, which is crucial for zero-example recognition.

Semantic Representations. To obtain a semantic representation for videos, inspiration is obtained from describing images with attributes [17], [45], and the video is represented by its attribute predictions. Creating the training data for a set of task-specific attribute classifiers manually involves lots of annotation effort, which is restrictive. Therefore, often public available datasets, such as ImageNet [30] and TRECVID [11], are used to train the attribute classifiers [23], [24], [25]. Although this overcomes the need for (additional) manual annotation, the attributes of these datasets are not necessarily descriptive for event recognition.

To tune the attributes for the task at hand, several works aim to automatically discover the attributes from web images/videos and their textual descriptions [31], [32], [33], [34]. For example Wu et al. [31], start by selecting the most frequent/relevant terms from a set of provided event descriptions as attributes. Then, inspired by Berg et al. [35], they use Internet search engines to gather positive examples. They assure visual predictability of the discovered attribute detectors by cross-validation. Similarly the work of Ye et al. [33] relies on WikiHow to obtain event descriptions, and the visual predictability of the selected terms is ensured by keeping only those terms which are present in existing image classification datasets. We refer to these methods as *term attributes* since they all discover the attributes from the terms in the descriptions.

Despite their effectiveness in obtaining training data, term attributes suffer from three drawbacks. First, descriptions have a long-tail distribution, therefore many terms rarely occur and these will not have enough positive examples to train reliable visual classifiers. Second, term attributes are selected mostly based on their visual prediction accuracy, while for effective event recognition the selected term attributes should also be descriptive for the target events. Third, contextual information is lost, since term attributes are learned independently by binary classifiers.

In this paper, we also learn the representation from videos and their descriptions. But, rather than selecting individual, and often unreliable, term attributes, we learn the entire representation by automatically combining the terms through embedding. In our embedding, the correlations between the terms are utilized to learn a more effective representation, which is predictable *and* descriptive.

2.2 Embedding Videos and Their Descriptions

To learn correspondences between the visual domain and textual descriptions different embedding methods have been proposed for various purposes, i.e., image annotation [46], image classification [47], [48], image captioning [49], [50], video to text translation [51], [52] and cross-modal retrieval [53], [54], [55].

Canonical correlation analysis (CCA) [56] is the classical unsupervised manner to relate different modalities and can be used for cross-modal retrieval [53]. CCA is the multi-modal generalization of PCA, and computed as a generalized eigenvalue problem on the cross-covariance matrix between the visual and textual features. It finds a sequence of uncorrelated projections in which the cross-correlation between the modalities is maximized. This is not necessarily suited to obtain a discriminative representation, as is also validated by our experiments.

Probabilistic topic models, such as the seminal correspondence latent Dirichlet allocation [46] and its extensions [57], [58], learn correspondences by extracting a set of correlated visual and textual topics from imagery and their captions. Despite their effectiveness for captioning images and videos [51], these methods are not designed to learn a representation for the purpose of recognition. Moreover, we note that by design these models are only applicable on discrete features, and therefore they cannot leverage the state-of-the-art video features used in our paper.

Embeddings for multi-class image classification, such as [47], [48] find a low-dimensional subspace in which multi-class classification is performed. The subspace is found jointly with the multi-class classifier by minimizing a classification loss [47], or a zero-shot classification loss [48]. In contrast to this multi-class image classification setting, we are interested in finding descriptions modeled as a multi-label video classification problem. Moreover, we argue that solely optimizing for classification is not sufficient to obtain a predictable and descriptive video embedding.

Recently deep neural network architectures have been proposed to learn multimodal correspondences for retrieval [54], [55] and captioning [49], [50]. Notably the multimodal recurrent neural networks capture the sequential ordering between the words in image captions and generate more accurate image captions, compared to the probabilistic topic models. However, training deep neural networks generally requires large amounts of training examples, while our purpose is to learn from few- and zero-examples.

2.3 Zero-Example Event Recognition

In zero-example event recognition the goal is to recognize an event based on a given textual *event definition*, without using any video examples. The event definition is usually provided in the form of a title and description, see Fig. 1. This zero-example setting is beyond the conventional zero-shot image recognition of objects and scenes [17], [18], [59], [60], where often a training set of related classes is available as well as pre-specified class-to-attribute mappings. This challenging event recognition problem has attracted attention because of its high practical value and the corresponding TRECVID benchmark task initiated by NIST [11]. The common approach is to represent videos and the event queries using a semantic representation, and to rank all videos based on the cosine similarity to the event query.

To represent the video mostly attributes and term-attributes [31], [32], [33], [34], [61], [62] are exploited. Extensions include combining attributes using logical operators [63], using video-segments to detect attributes [64], and adjustment of attribute scores using an ontology structure [29]. In addition to attributes, automatic speech recognition and optical character recognition have been considered to enrich the semantic video representations [29], [31], [65].

To answer the event query, usually term matching between the semantic video representation and the event definition is performed. The retrieval can be enriched by using contextual information, such as word embeddings and term co-occurrences [34], [66]. For the final ranking, the cosine similarity can be extended by pseudo relevance feedback mechanisms, such as self-paced ranking [62]. This has been used to improve the zero-example event recognition by re-ranking [29], [61], [66]. Our representation learning is orthogonal to these efforts and can be joined with them to further improve the event recognition performance.

3 Video2vec Embeddings

Our goal is to learn a representation function $f : \mathcal{X} \rightarrow \mathcal{S}$, which maps each low-level video feature $\boldsymbol{x}_i \in \mathcal{X}$ into the semantic representation $\boldsymbol{s}_i \in \mathcal{S}$. The representation function is trained on a collection of videos and their semantic descriptions. We represent the descriptions in the form of binary term vectors $\boldsymbol{y}_i \in \mathcal{Y}$. For event recognition, the $\boldsymbol{s}_i$ should have two properties: First, it should be descriptive enough to provide a characteristic representation of each video. Second, each dimension of the representation, corresponding to a semantic concept, should be predictable from the low-level video features.

In the attribute approach to learning the representation, the semantic representations are defined to be in the same space as the term vectors ($\mathcal{S} = \mathcal{Y}$). However, in practice, the term vectors are noisy and sparse, which undermines their effectiveness as labels to train f. Therefore, we propose to learn the $\boldsymbol{s}_i$ on a lower dimensional projection of the term vectors, which are less sparse and less noisy. To remedy that the lower dimensional term vectors may be less descriptive for the video content, we formulate f as an embedding, coined as Video2vec, which is learned by balancing the descriptiveness and predictability as a joint optimization.

We first introduce the Video2vec embedding and its application for few- and zero-example event recognition (Section 3.1). Then we generalize our embedding to fuse multimedia features (Section 3.2). Finally, we extend the embedding to be learned specifically per event (Section 3.3). We summarize our notation conventions in Table 1.

3.1 Objective Function

We start from a dataset of videos, represented by video features $\boldsymbol{X}$, and their textual descriptions, represented by binary term vectors $\boldsymbol{Y}$, indicating which words are present in each video description. Then, our Video2vec representation is learned by minimizing

$$L_{\mathrm{V}}(\boldsymbol{A}, \boldsymbol{W}) = \min_{\boldsymbol{S}} L_d(\boldsymbol{A}, \boldsymbol{S}) + L_p(\boldsymbol{S}, \boldsymbol{W}), \tag{1}$$

TABLE 1
Summary of the Core Notation Used for Video2vec

Notation	Description
N	Number of videos
M	Number of unique words in descriptions
D	Dimensionality of low-level feature
J	Number of low-level features to fuse
k	Dimensionality of Video2vec embedding
$\boldsymbol{X} \in \mathbb{R}^{D \times N}$	Matrix of low-level video features
$\boldsymbol{Y} \in \{0,1\}^{M \times N}$	Matrix of binary term vectors
$\boldsymbol{W} \in \mathbb{R}^{D \times k}$	Video2vec visual projection
$\boldsymbol{A} \in \mathbb{R}^{M \times k}$	Video2vec textual projection
$\boldsymbol{S} \in \mathbb{R}^{k \times N}$	Video2vec embedding
$\boldsymbol{H} \in \mathbb{R}^{M \times M}$	Diagonal matrix with per-term weights
$\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{s}_i$	The column representing the ith video

where $\boldsymbol{A}$ is the textual projection matrix, $\boldsymbol{W}$ is the visual projection matrix, and $\boldsymbol{S}$ is the Video2vec embedding. The loss function L_d corresponds to our first objective for learning a descriptive Video2vec, and the loss function L_p corresponds to our second objective for learning a predictable Video2vec. The Video2vec embedding $\boldsymbol{S}$ interconnects the two loss functions.

Descriptiveness. For the L_d function, we use a variant of regularized Latent Semantic Indexing (LSI) [67]. This objective minimizes the quadratic error between the original video descriptions $\boldsymbol{Y}$, and the reconstructed translations obtained from $\boldsymbol{A}$ and $\boldsymbol{S}$

$$L_d(\boldsymbol{A}, \boldsymbol{S}) = \frac{1}{2} \sum_{i=1}^{N} \|\boldsymbol{y}_i - \boldsymbol{A}\boldsymbol{s}_i\|_2^2 + \lambda_a \Omega(\boldsymbol{A}) + \lambda_s \Psi(\boldsymbol{S}), \quad (2)$$

where $\Psi(\cdot)$ and $\Omega(\cdot)$ denote regularization functions, and $\lambda_a \geq 0$ and $\lambda_s \geq 0$ are regularizer coefficients. We use the squared Frobenius norm for regularization, which is the matrix variant of the ℓ_2 regularizer, i.e., $\Omega(\boldsymbol{A}) = \frac{1}{2}\|\boldsymbol{A}\|_{\mathrm{F}}^2 = \frac{1}{2}\sum_i \|\boldsymbol{a}_i\|_2^2 = \frac{1}{2}\sum_{ij} a_{ij}^2$, the sum of the squared matrix elements. Similarly for the Video2vec matrix $\Psi(\boldsymbol{S}) = \frac{1}{2}\|\boldsymbol{S}\|_{\mathrm{F}}^2$.

The main difference with regularized Latent Semantic Indexing [67] is their ℓ_1 regularizer, $\Omega(\boldsymbol{A}) = \sum_i \|\boldsymbol{a}_i\|_1$, which enforces sparsity in the textual projection $\boldsymbol{A}$. However, with our larger representation (typically we use a dimensionality of k between 512 and 2,048 in our experiments compared to only $k = 20$ in [67]) and lower number of unique words (around 10 K, compared to 100 K), enforcing sparsity is not necessary for good performance.

Note that other textual embeddings can be formulated similar to Eq. (2), when appropriate regularization functions $\Omega(\cdot)$ and $\Psi(\cdot)$ are used. For example using $\Omega(\cdot) = \|\cdot\|_1$ enforces sparsity [67], [68], [69], or in the extreme case that $\boldsymbol{A}$ is constrained such that each column has a single non-zero value, the objective becomes very close to methods that select the best single term labels [35], and when $\boldsymbol{A}$ is enforced to preserve the taxonomical term relations, the objective resembles taxonomy embedding [70], [71].

Predictability. The L_p function measures the occurred loss between the Video2vec $\boldsymbol{S}$ and the embedding of video features using $\boldsymbol{W}$. Since the Video2vec $\boldsymbol{S}$ is real valued, as opposed to a binary or multi-class encoding, we can not rely on standard classification losses such as the hinge-loss used in SVMs. Therefore, we define L_p as a regularized regression, similar to ridge regression

$$L_p(\boldsymbol{S}, \boldsymbol{W}) = \frac{1}{2} \sum_{i=1}^{N} \|\boldsymbol{s}_i - \boldsymbol{W}^\top \boldsymbol{x}_i\|_2^2 + \lambda_w \Theta(\boldsymbol{W}), \quad (3)$$

where we use (again) the Frobenius norm for regularization of the visual projection matrix W, $\Theta(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{W}\|_{\mathrm{F}}^2$, and λ_w is the regularization coefficient.

Joint Optimization. To handle large scale datasets and state-of-the-art high-dimensional visual features, e.g., Fisher vectors [44] on video features [12] or deep learned representations [42], we employ a Stochastic Gradient Descent (SGD) [72] optimization, summarized in Algorithm 1. The number of passes over the datasets (*epochs*) m and the step-size η are hyper-parameters of SGD.

Algorithm 1. Pseudocode for Learning the Video2vec

input : $\boldsymbol{X}$, $\boldsymbol{Y}$, k, η (step-size), m (max-epochs)
output : $\boldsymbol{W}$ and $\boldsymbol{A}$
$\boldsymbol{A}$, and $\boldsymbol{S} \leftarrow$ SVD decomposition of $\boldsymbol{Y}$
$\boldsymbol{W} \leftarrow$ random (zero-mean)
for $e \leftarrow 1$ **to** m **do**
  for $i \leftarrow 1$ **to** N **do**
    Pick a random video-description pair $(\boldsymbol{x}_t, \boldsymbol{y}_t)$
    Compute gradients $w.r.t.$ $\boldsymbol{A}, \boldsymbol{W}$ and $\boldsymbol{s}_t$
    Update parameters:

$$\begin{aligned} \boldsymbol{A} &\leftarrow \boldsymbol{A} - \eta_t \nabla_{\boldsymbol{A}} L_{\mathrm{V}} && \text{see Eq. (4)} \\ \boldsymbol{W} &\leftarrow \boldsymbol{W} - \eta_t \nabla_{\boldsymbol{W}} L_{\mathrm{V}} && \text{see Eq. (5)} \\ \boldsymbol{S} &\leftarrow \boldsymbol{s}_t - \eta_t \nabla_{\boldsymbol{s}_t} L_{\mathrm{V}} && \text{see Eq. (6)} \end{aligned}$$

  end
end
return: $\boldsymbol{W}$ and $\boldsymbol{A}$

The Video2vec objective function, as given in Eq. (1), is convex with respect to matrix $\boldsymbol{A}$ and $\boldsymbol{W}$ when the embedding $\boldsymbol{S}$ is fixed. In that case, the joint optimization is decoupled into Eqs. (2) and (3), which are both reduced to a standard ridge regression for a fixed $\boldsymbol{S}$. Moreover, when both $\boldsymbol{A}$ and $\boldsymbol{W}$ are fixed, the objective in Eq. (1) is convex $w.r.t.$ $\boldsymbol{S}$. Therefore we use standard SGD by computing the gradients of a sample $w.r.t.$ the current value of the parameters, and we minimize $\boldsymbol{S}$ jointly with $\boldsymbol{A}$ and $\boldsymbol{W}$.

Lets denote a randomly sampled video and description pair at step t by $(\boldsymbol{x}_t, \boldsymbol{y}_t)$, and let $\boldsymbol{s}_t$ denote the current Video2vec embedding of sample t. The gradients of Eq. (1) for this sample $w.r.t.$ $\boldsymbol{A}, \boldsymbol{W}$ and $\boldsymbol{s}_t$ are given by

$$\nabla_{\boldsymbol{A}} L_{\mathrm{V}} = -\big(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{s}_t\big)\boldsymbol{s}_t^\top + \lambda_a \boldsymbol{A}, \quad (4)$$

$$\nabla_{\boldsymbol{W}} L_{\mathrm{V}} = -\,\boldsymbol{x}_t\big(\boldsymbol{s}_t - \boldsymbol{W}^\top \boldsymbol{x}_t\big)^\top + \lambda_w \boldsymbol{W}, \text{ and} \quad (5)$$

$$\nabla_{\boldsymbol{s}_t} L_{\mathrm{V}} = -\,\boldsymbol{A}^\top\big(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{s}_t\big) + \big(\boldsymbol{s}_t - \boldsymbol{W}^\top \boldsymbol{x}_t\big) + \lambda_s \boldsymbol{s}_t. \quad (6)$$

The effect of joint learning descriptiveness and predictability, becomes clear in Eq. (6), where both the textual projection matrix $\boldsymbol{A}$ and visual projection matrix $\boldsymbol{W}$ contribute to learning the Video2vec embedding $\boldsymbol{S}$. This embedding $\boldsymbol{S}$ is subsequently used to obtain the textual projection $\boldsymbol{A}$ matrix, in Eq. (4), and the visual projection $\boldsymbol{W}$ matrix, in Eq. (5).

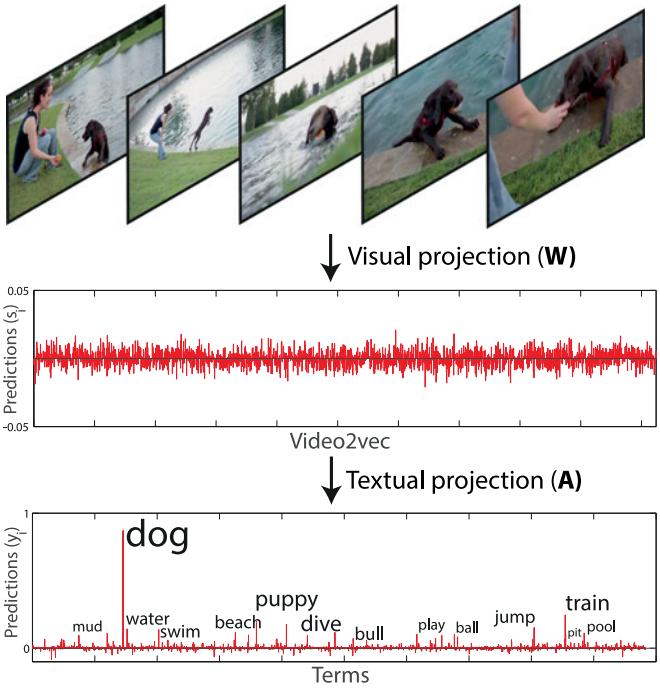

Fig. 2. Video2vec prediction: From the low-level video features the Video2vec representation and the term vector are predicted using the visual projection matrix W and the textual projection matrix A.

This leads to the Video2vec embedding, which is both descriptive, by preserving the textual information, and predictable, by minimizing the visual prediction loss.

Initialisation. The parameters $\boldsymbol{A}$, $\boldsymbol{S}$, and $\boldsymbol{W}$ can be initialized randomly. However, to speed up convergence, we propose to initialize $\boldsymbol{A}$ and $\boldsymbol{S}$ by a rank-reduced singular value decomposition (SVD) of term vectors $\boldsymbol{Y}$. This corresponds to the solution of the unregularized LSI objective of Eq. (2). For this purpose, $\boldsymbol{A}$ and $\boldsymbol{S}$ are initialized by the k first eigenvectors of $\boldsymbol{Y}\boldsymbol{Y}^\top$ and $\boldsymbol{Y}^\top\boldsymbol{Y}$, respectively.

Predicting Video2vec. After training the visual and textual projection matrices, they are used to predict the Video2vec representations and the term vectors. In the case that both a video $\boldsymbol{x}_i$ and description $\boldsymbol{y}_i$ are given, we could obtain the Video2vec representation by returning $\boldsymbol{s}_i$ from Eq. (1), while keeping both $\boldsymbol{A}$ and $\boldsymbol{W}$ fixed. However, in practice most videos lack a description. Therefore, we use

$$\boldsymbol{s}_i = \boldsymbol{W}^\top \boldsymbol{x}_i, \tag{7}$$

to predict our Video2vec representation from the low-level video features $\boldsymbol{x}_i$. Moreover, using the predicted representation $\boldsymbol{s}_i$, the term vectors for each unseen video are predicted

$$\hat{\boldsymbol{y}}_i = \boldsymbol{A}\boldsymbol{s}_i = \boldsymbol{A}\boldsymbol{W}^\top \boldsymbol{x}_i, \tag{8}$$

where the words with the highest values are most relevant for this video, see the illustration in Fig. 2.

Zero-Example Event Recognition. To enable zero-example recognition using the Video2vec embedding, we employ the following steps: First, each test video is represented by predicting its term vector $\hat{\boldsymbol{y}}_i$ using Eq. (8), based on the pretrained embeddings. Second, we translate the textual event definition into the event query, denoted as $\boldsymbol{y}^e \in \mathbb{R}^M$, by matching the words in the event definition with the M unique words in the Video2vec dictionary. Finally, the zero-example ranking is obtained by measuring the similarity between the video representations and the event query based on the cosine similarity

$$s_e(\boldsymbol{x}_i) = \frac{\boldsymbol{y}^{e\top}\hat{\boldsymbol{y}}_i}{||\boldsymbol{y}^e|| \quad ||\hat{\boldsymbol{y}}_i||}. \tag{9}$$

3.2 Video2vec Fusion

Videos are inherently multimodal. In general any video contains appearance, motion, and audio cues and sometimes even textual information in the form of subtitles or speech recognition scripts. Fusing the different modalities is typically achieved by early-fusion, i.e., fusion at the level of the representations, and late-fusion, i.e., fusion at the level of prediction scores [73]. Both fusion strategies have been shown to be effective for understanding complex events as well, e.g., [4], [31], [41], [74]. We propose *Video2vec*$^{\mathcal{F}}$, which extends the Video2vec embedding by learning the semantic representation from multiple modalities.

A straightforward approach to learn the multimodal semantic representation is by fusing multiple Video2vec embeddings, which are independently trained per modality. Our intuition is that the semantic representation is more effective if it is predictable from *all* the modalities rather than from each individual modality. For *Video2vec*$^{\mathcal{F}}$ we adjust the predictability loss, of Eq. (3), to incorporate a weighted combination of the predictability from all J modalities as

$$L_p^{\mathcal{F}}(\boldsymbol{S}, \mathcal{W}) = \sum_{j=1}^{J} L_p(\boldsymbol{S}, \boldsymbol{W}^j) \tag{10}$$

where $\boldsymbol{S}$ is the multimodal Video2vec embedding, and $\mathcal{W} = \{\boldsymbol{W}^j, j = 1 \ldots J\}$ is a set of projection matrices from the J modalities. Each feature projection matrix $\boldsymbol{W}^j \in \mathbb{R}^{D_j \times k}$ projects the low-level feature $\boldsymbol{x}_i^j \in \mathbb{R}^{1 \times D_j}$ extracted from the video into the Video2vec representation $\boldsymbol{s}_i$. To balance the impact of feature dimensionality in the multimodal predictability loss, all low-level features from $\boldsymbol{x}_i^j$ are ℓ_2 normalized. Moreover, instead of weighting each modality equally, as in Eq. (10), a term $\gamma_j \geq 0$ could be added to weight the importance of each modality, if sufficient training examples are available for their cross-validation.

The objective function Eq. (1) is still convex with respect to $\boldsymbol{S}$, $\boldsymbol{A}$, and $\boldsymbol{W}^j$ when the other parameters are fixed. However, the gradient with respect to $\boldsymbol{s}_t$, Eq. (6) becomes

$$\nabla_{\boldsymbol{s}_t} L_{\mathrm{V}} = -\boldsymbol{A}^\top(\boldsymbol{y}_t - \boldsymbol{A}\boldsymbol{s}_t) + (\boldsymbol{s}_t - \sum_j \boldsymbol{W}^{j\top}\boldsymbol{x}_t^j) + \lambda_s \boldsymbol{s}_t. \tag{11}$$

It can be seen that all the modalities are jointly contributing to learn the multimodal Video2vec representations $\boldsymbol{S}$.

After training the textual and feature projection matrices, they are used to extract the multimodal Video2vec representation. Each projection matrix $\boldsymbol{W}^j$ predicts the Video2vec representation based on its underlying modality as

$$\boldsymbol{s}_i^j = \boldsymbol{W}^{j\top}\boldsymbol{x}_i^j. \tag{12}$$

The final Video2vec representation is obtained by aggregating the per-modality representations. We experimentally observe that for few-example recognition, concatenation

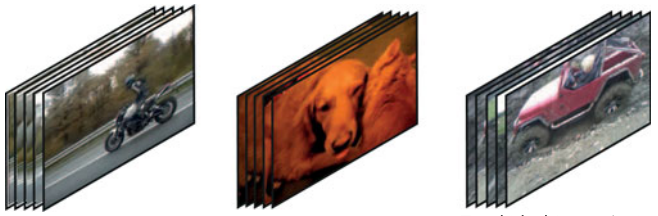

Fig. 3. Example videos and title captions from the VideoStory46K dataset [37], which we use for Video2vec representation learning.

performs slightly better than sum or max pooling. Hence we use a concatenation of per-modality embeddings as the final representation: $\boldsymbol{s}_i = \left[\boldsymbol{s}_i^1, \ldots, \boldsymbol{s}_i^J\right]$. For zero-example recognition, we predict the final term vector by summing up the per-modality term vectors: $\hat{\boldsymbol{y}}_i = \sum_{j=1}^{J} \boldsymbol{A}\boldsymbol{s}_i^j$.

3.3 Event Specific Video2vec

Not all words are equally important for recognizing an event. For each event there are a few terms which are highly informative, while the others are not. For the event *"attempting a bike trick"* words such as `bicycle`, `jumping`, and `ramp` are highly distinctive while `beach`, `cake`, and `children` are less informative.

In Section 3.1, the descriptiveness loss L_d is defined as the overall error in reconstructing *all* the words from the Video2vec representations, see Eq. (2). With this definition, the loss is biased towards the more frequent words, as minimizing their reconstruction error leads to a higher decrease in the overall error. Consequently, the words which are infrequent in the descriptions might be discarded, which degrades their prediction accuracy from video features. This undermines the effectiveness of our representation learning.

We extend the Video2vec embedding to learn a video representation, able to predict the informative words of events. Our extension, which we coin as *Video2vec*$_{\mathcal{E}}$, is learned per event. It minimizes the reconstruction error of the terms with respect to their importance for event description, rather than their frequency in the Video2vec train data. We introduce a *term sensitive* descriptiveness loss

$$L_d^{\text{ts}}(\boldsymbol{A}, \boldsymbol{S}) = \frac{1}{2}\sum_{i=1}^{N} \|\boldsymbol{H}_e^{\frac{1}{2}}(\boldsymbol{y}_i - \boldsymbol{A}\boldsymbol{s}_i)\|_2^2 + \lambda_a \Omega(\boldsymbol{A}) + \lambda_s \Psi(\boldsymbol{S}), \quad (13)$$

where $\boldsymbol{H}_e \in \mathbb{R}^{M \times M}$ is a diagonal matrix, denoting the importance of each word for describing an event. By setting a relatively high value for h_{jj} for term j, its reconstruction error is more penalized compared to the other terms. Hence the word is expected to be more precisely reconstructed.

We determine the term importance matrix $\boldsymbol{H}_e$ by relying on the presence/absence of terms in the textual event definitions. Our assumption is that the words, which are present in event definitions are more important than the absent words. We set each element of the importance matrix h_{jj} to α, if the word j is present, and $1 - \alpha$ if the word j is absent in the event definition. α is a balancing parameter between 0 and 1, which should be higher than 0.5 to assign more importance to the present words. We empirically set this parameter to 0.75 in all our zero-example experiments.

Multimodal Fusion. Finally, we can leverage the multimodal features for learning event specific embeddings by combining the multimodal predictability loss and the term sensitive descriptiveness in a joint objective

$$L_{\text{V}}(\boldsymbol{A}, \boldsymbol{W}) = \min_{S} L_d^{ts}(\boldsymbol{A}, \boldsymbol{S}) + L_p^{\mathcal{F}}(\boldsymbol{S}, \boldsymbol{W}). \quad (14)$$

We coin the learned video representation *Video2vec*$_{\mathcal{E}}^{\mathcal{F}}$, since it is learned event specific on multiple video modalities.

4 Experimental Setup

4.1 Datasets

We first introduce the dataset used for learning the Video2vec embeddings. Then, we detail the video datasets by which we evaluate the event recognition experiments.

4.1.1 Video2vec Learning

In all experiments, we learn the Video2vec embeddings on the *VideoStory46K* [37] dataset. This collection encompass 45,826 videos harvested from YouTube, with a total length of 743 hours. Every video comes with a short title caption provided by the user who has uploaded the video, as shown in Fig. 3. There are 19,159 unique terms in the captions, most of them occurring infrequently. We filter out the terms occurring only once as they generally are misspelled terms, numbers, or noisy terms. It provides us with 9,828 unique terms, which are used in our experiments.

4.1.2 Event Recognition Evaluation

We perform our event recognition on the challenging TRECVID Multimedia Event Detection (MED) corpus [75] and the Columbia Consumer Video (CCV) collection [76]. These contain more than 42 K videos in total, including user generated web videos with a large variation in quality, length and content.

TRECVID Multimedia Event Detection [75]. This dataset is introduced by NIST as a benchmark for event recognition. We perform our experiments on the two latest releases of the dataset: *MED 2013* and *MED 2014*. Each dataset includes videos from 20 complex events, with 10 overlapping as listed in Table 5. Each dataset includes three labeled video partitions: Event Kit training, Background training, and test set MED including 200, 5 K, and 27 K videos, respectively.[1] Apart from the videos, a textual definition is provided per event, which explicates the event as unformatted plain text, such as the one shown in Fig. 1.

We perform our few- and zero-example experiments by exactly following the *10Ex* and *0Ex* evaluation procedure outlined by the NIST TRECVID event recognition task [11]. In the few-example experiments, training data for each event is composed of 10 positive videos from the Event Kit training data along with about 5 K negative videos from the Background training data. The results for each event classifier are reported on the 27 K videos from test set MED. In our zero-example experiments, we rely on the provided textual event definitions to create an event query vector. Then the performance is reported on the test set MED.

1. There is also a PROGRESS set with 98 K videos, but this partition is for blind testing by NIST only.

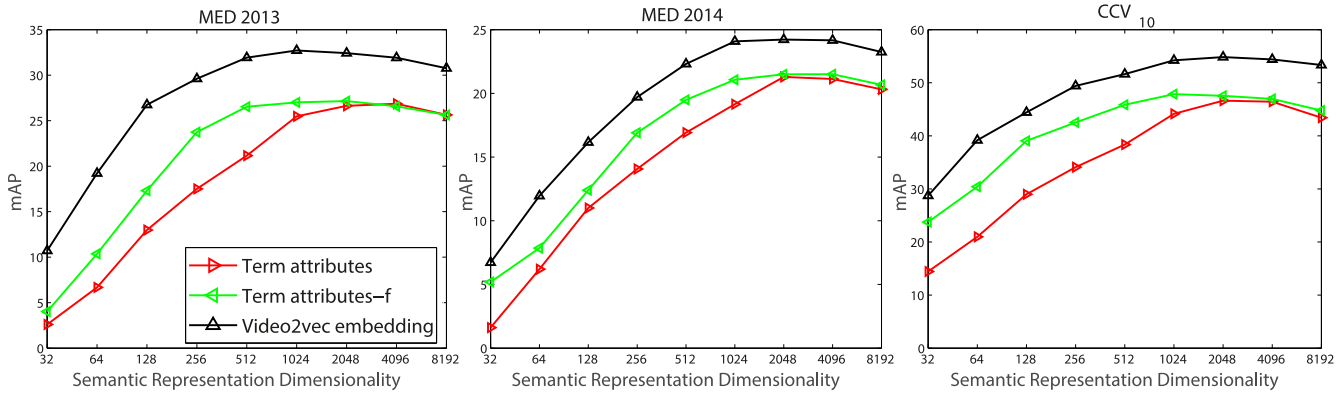

Fig. 4. Effect of embedding. The Video2vec embedding outperforms the term attribute and term attribute-f baselines, which are directly learned from the terms without embedding.

Columbia Consumer Video [76]. This dataset contains 9,317 user-generated videos from YouTube including over 210 hours in total. The dataset contains ground truth annotations at video-level for 20 semantic categories, i.e., *wedding reception* and *music performance*. We use the standard partitioning of the dataset, but we use only 10 positive examples per event in the training data. These 10 are selected based on alphabetical order of the respective video names, we ignore the remaining positive examples in the train set. We report event recognition results on the standard test partition. We denote our redefinition of the CCV dataset for few-example event recognition as $\boldsymbol{CCV}_{10}$.

4.2 Video Features

To cancel out accidental effects of the choice for the underlying features, we consider the same set of appearance, motion and audio features for all our experiments, the various baselines, and our Video2vec variants. All the experiments are performed using the appearance features. In Sections 5.2, 5.3, and 5.4, the motion and audio features are also included for the multimodal fusion experiments.

Appearance. We adopt the *video CNN* representation [1] as appearance features for event recognition, but found the very deep network of [38] to perform slightly better than [9]. For each video the frames are extracted by uniformly sampling every two seconds. Then, the CNN descriptors are extracted per frame as the 1 K dimensional responses from the last fully connected layer (pool_5) of the Google Inception network [38]. We train the network on the 15,293 ImageNet categories with more than 200 examples, using the Caffe toolbox [77]. The final video CNN features are obtained by aggregating the frame descriptors over each video by VLAD encoding [10] with a codebook size of 20, resulting in a 20,480 dimensional vector.

Motion. We use the MBH descriptors along the motion trajectories [12] as motion features. The extracted 288-dimensional descriptors are reduced to 128 dimensions using PCA and are then aggregated per video using a Fisher vector [44], with 128 Gaussians resulting in a 32,768 dimensional vector. Each Fisher vector is power normalized, with $\alpha = 0.2$, as in [78].

Audio. We extract MFCCs descriptors [79] over a 10 ms window. The descriptors consist of 13 values, 30 coefficients and the log-energy, along with their derivatives and the second derivatives. The MFCC descriptors are aggregated by Fisher vectors using a Gaussian Mixture Model with 256 components, resulting in a 46,080 dimensional vector.

4.3 Implementation Details

We learn the Video2vec embeddings by using 75 percent of the VideoStory46K dataset for training and 25 percent for validation to set the hyper-parameters of our model ($\lambda_w, \lambda_a, \lambda_s$) and of SGD (number of epochs, η). As the validation criterion we rely on the objective function value, when using $S = W^\top X$. The step size η_t is fixed during training.

For few-example event recognition, the event classifiers are trained as binary SVM with RBF kernels, as suggested in [19]. Following [1], [15] we set the SVM regularization and the RBF kernel parameters by a default value of 1, as the train set is not big enough for cross-validation.

As evaluation criteria we follow the standard convention in the literature [11], [76] by relying on the average precision (AP) per event, and we report the mean average precision (mAP) for overall accuracy.

5 Experiments

5.1 Video2vec for Few-Example Event Recognition

5.1.1 Effect of Embedding

We first compare Video2vec with term attribute baselines. The baselines learn the representations directly from terms without any embedding. We evaluate all video representations for few-example event recognition using a dimensionality varying from 32 to 8,192.

1. *Term attributes.* This representation is extracted by following the tradition of predicting relevant individual words from the video descriptions, e.g., [31], [32], [33], [34]. A linear SVM classifier is trained per term. The classifiers which have the highest prediction accuracy, based on a two-fold cross-validation [35], are selected as term attributes.
2. *Term attributes-f.* Rather than using cross-validation to select the term attributes it simply selects the words with the highest *frequency* in the descriptions.

Results. The results in Fig. 4 demonstrate that Video2vec embeddings outperform the term attribute and term attribute-f baselines on all the three test sets.

Term attributes, which relies on the estimated reliability of individual word classifiers suffers from two drawbacks.

TABLE 2
Five Selected Dimensions of the Video2vec Embedding Trained on VideoStory46K

car	diamond	cute	cell	room
wheel	style	cat	mobile	living
man	pear	play	home	designer
truck	set	dog	call	furniture
test	gold	meow	solavei	picture
woman	engagement	baby	plan	fedisa
drive	body	kitty	business	car
front	heart	cutest	girl	truck
driver	dress	black	woman	traffic
rack	card	bath	man	fire

Each dimension is visualized by reporting its 10 most relevant words, as determined by $\boldsymbol{A}^{-1}$. By design, correlated terms are usually combined into one dimension of Video2vec.

First, many of the visual terms are very specific and therefore incapable of characterizing the events of interest, i.e., `necklace`, `suitcase`, and `earring`. Although these words can be accurately predicted from videos, they are incapable of providing a characteristic representation for event recognition. Second, many of the words rarely occur in video descriptions. Hence, only a limited number of positive examples are available to learn the word classifiers, which leads to a biased estimation of their reliability. Consequently, many of the discovered visual terms overfit to their small training set and do not generalize well for new videos.

The drawbacks of term attributes are relaxed by simply relying on the most frequent words. We observe the most frequent terms usually refer to characteristic attributes of events which are frequently used by humans when describing a video, i.e., `car`, `girl`, and `kid`. Because of their large number of positive examples, the trained visual classifiers from term attributes-f are in general more reliable.

The Video2vec embedding represents the words in a reduced-dimensional space, where correlated terms are usually combined together, as visualized in Table 2. Combining correlated words leads to less correlation between dimensions of the learned representation. Moreover, as the positive examples for all correlated terms are combined, it provides more positive video examples to train visual classifiers, often leading to better accuracy.

Besides its effectiveness for few-example event recognition, the Video2vec embedding also improves the representation learning efficiency by training visual embeddings for combination of words rather than individual words.

5.1.2 Video2vec versus Other Embeddings

We compare the effectiveness of our proposed Video2vec embedding with two alternative embeddings:

1. *CCA embedding*. This baseline learns the textual and visual projections by CCA [53], which maximizes the cross-correlation between the video features and descriptions. We experimentally observed that the embedding is even more effective when we PCA-reduce the video features to a dimensionality of 1,024 before learning the CCA embedding.
2. *Description embedding*. Similar to the Video2vec, this embedding is learned by minimizing the descriptiveness and predictability losses, but in two disjoint steps: The textual projection is first learned based on the regularized Latent Semantic Indexing [67], as in Eq. (2). Then the visual projection is learned separately, by minimizing the error for predicting the embedded descriptions from the video features based on ridge regression, as in Eq. (3).

Results. The results in Fig. 5 demonstrate that the Video2vec embedding outperforms the CCA and the Description embedding baselines on all the three test sets for a dimensionality larger than 256.

We explain the gain over CCA by the fact that CCA is a symmetric embedding, it learns both the textual and visual projections in the same way. However, the textual and visual features have different distributions and properties, which may require different objective functions to learn their projections. This is achieved by Video2vec as it relies on two separate LSI and ridge regression loss functions to learn the textual and visual projections.

We explain the improvement over the Description embedding by the fact that combining the words based on textual correlation only does not necessarily imply that the corresponding video is visually correlated as well. As it happens, the term pairs `puppy` and `kid`, `cake` and `dance`, and `car` and `fire` have high correlations in the descriptions but are visually dissimilar. Combining these

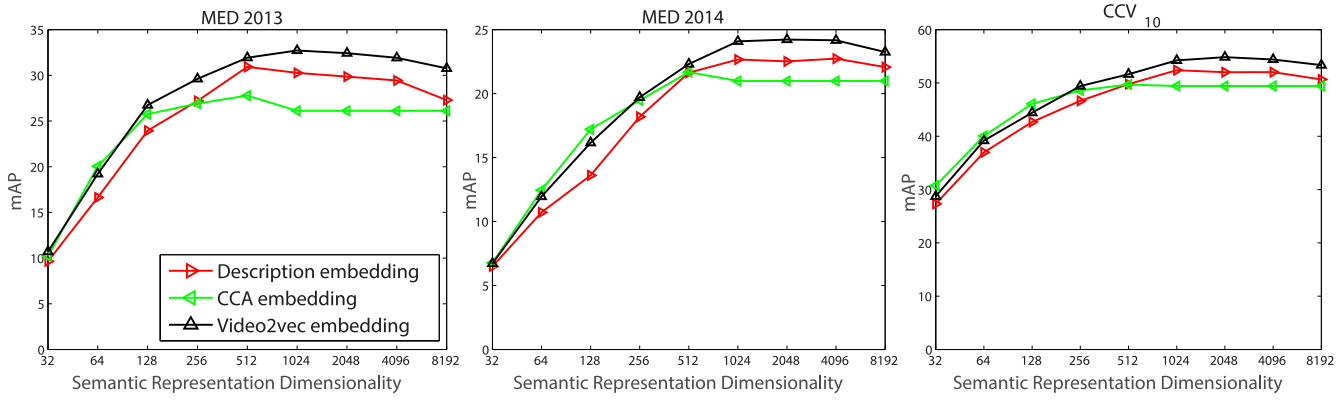

Fig. 5. Video2vec versus other embeddings. Video2vec outperforms the CCA and the description embedding on all three test sets. The description embedding is the closest competitor, but it suffers from embedding correlated terms which are visually dissimilar. CCA, uses the same objective function to learn the visual and textual embeddings, which is suboptimal due to intrinsic differences between the visual and textual features.

TABLE 3
Video2vec versus Other Representations for Few-Example Event Recognition

	Low-Level	Attributes	Video2vec
MED 2013	28.1	22.5	**32.4**
MED 2014	21.8	17.2	**24.2**
CCV_{10}	50.0	48.8	**54.8**

Video2vec outperforms the alternatives on all three test sets.

words together, as is done by Description embedding, undermines the accuracy of the classifiers predicting them from videos. In contrast, in a Video2vec the correlated words are combined only if their combination improves their classifier prediction. It penalizes the combination of correlated terms which are visually dissimilar.

From now on we fix the Video2vec representation to 2,048 dimensions, which is optimal based on Fig. 4.

5.1.3 Video2vec versus Other Representations

Next, we compare Video2vec with state-of-the-art video representations for few-example event recognition:

1. *Low-Level.* In this baseline, the event classifiers are trained directly on the low-level video representations, without extracting a semantic representation. We rely on the video CNN features of [1], based on our implementation detailed in Section 4.2.
2. *Attributes.* This representation is obtained by adopting the public ImageNet dataset as the source for training attribute classifiers as proposed in [23]. However, instead of training SVM classifiers on bag-of-words encoding of color SIFT descriptors [23], we upgrade the attributes by training a deep CNN with Google Inception architecture on the 15,293 ImageNet categories as detailed in Section 4.2.

Results. Table 3 shows that Video2vec outperforms the state-of-the-art attributes and low-level video representations on all three test sets.

By comparing the Video2vec and the attribute representation we observe a higher event recognition accuracy of 32.4 versus 22.5 for the MED 2013, 17.2 versus 21.8 for the MED 2014, and 54.8 versus 48.8 for the CCV_{10} test set. We explain it by the reliance of the attribute baseline on ImageNet categories. Many of these pre-specified categories are not semantically relevant for the events of interest. For example, many of the categories are devoted to specific animal species. In contrast, the Video2vec embedding is automatically derived from the VideoStory46K dataset, which includes many descriptions relevant to events. Different from previous work, our low-level representation outperforms the attribute representation, indicating the strong low-level features used in this work.

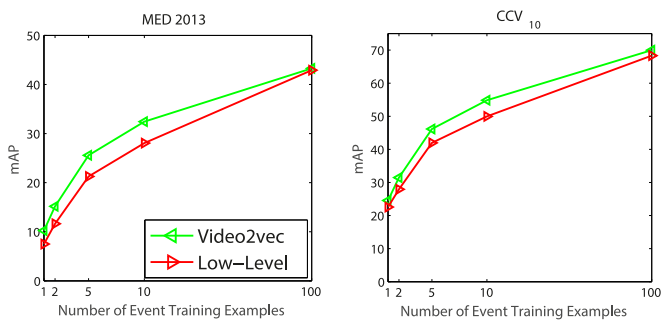

Fig. 6. From 1 to 100 examples. When the number of event exemplars is limited Video2vec outperforms the low-level representation. By increasing training examples the difference becomes more subtle.

TABLE 4
Video2vec versus Other Semantic Representations for Zero-Example Event Recognition

	Attributes	Term-attributes	Term-attributes-f	Video2vec
MED 2013	3.2	10.7	14.2	**15.9**
MED 2014	1.4	3.8	5.0	**5.2**

Our proposed Video2vec outperforms the alternatives.

The results further demonstrate that the Video2vec outperforms the low-level representation. As an explanation, we speculate that the low-level representation is prone to overfitting due to its high dimensionality. More specifically, in the low-level baseline, the event classifiers are trained on the 20,480 dimensional low-level features from only 10 positive exemplars. This may lead to overfitting as a result of the curse of dimensionality. In contrast, the Video2vec representation transfers the semantics from descriptions into the video representation to alleviate the overfitting as a sort of regularization.

From 1 to 100 Examples. For further investigation, we gradually increase the number of positive examples from 1 to 100, and evaluate the accuracy of the event classifiers trained on both the low-level and Video2vec representation. The positive examples are selected randomly[2] and the results are reported by averaging over 10 repetitions to compensate for the random effect. As Fig. 6 shows, when the number of event train examples is less than 10, the Video2vec representation outperforms the low-level representation. By increasing the training examples to 100, the difference becomes more subtle, which confirms our hypothesis.

5.1.4 Video2vec for Zero-Example Event Recognition

We also evaluate the effectiveness of Video2vec for zero-example event recognition. We compare Video2vec with the three semantic representation baselines used before: *Attributes*, *Term attributes*, and *Term attributes-f*.

Results. The results in Table 4 demonstrate that Video2vec outperforms the baselines on both test sets. We again explain the modest performance of the attribute baseline by the fact that this representation relies on ImageNet categories as attributes. Many of which are not semantically relevant to the events of interest. This drawback is relaxed by the term attributes and term-attributes-f baselines, as they are trained on more relevant video descriptions from the VideoStory46K dataset. However, both term attribute baselines rely on individual terms for learning the representation. In contrast, Video2vec learns the representation from term combinations, which leads to a better zero-example event recognition accuracy.

2. The additional event exemplars are selected from the 100Ex evaluation procedure provided in the MED dataset.

TABLE 5
Effect of Fusion

MED 2013				MED 2014				CCV_{10}			
Event	Appearance	+ Motion	+ Audio	*Event*	Appearance	+ Motion	+ Audio	*Event*	Appearance	+ Motion	+ Audio
Birthday party	37.1	38.8	**43.4**	Attempting a bike trick	9.0	**9.4**	8.8	Basketball	65.6	67.9	**68.6**
Changing a vehicle tire	64.7	65.5	**67.2**	Cleaning an appliance	11.1	13.7	**16.0**	Baseball	58.6	**59.6**	59.5
Flash mob gathering	55.3	63.8	**65.0**	Dog show	83.7	88.0	**89.7**	Soccer	58.7	64.5	**64.7**
Getting a vehicle unstuck	59.3	65.2	**65.3**	Giving directions to a location	0.5	**1.4**	1.0	Ice Skating	70.2	74.5	**75.1**
Grooming an animal	24.3	**29.4**	28.2	Marriage proposal	0.3	0.5	**0.6**	Skiing	79.4	81.8	**81.9**
Making a sandwich	16.7	18.9	**21.3**	Renovating a home	11.5	11.9	**12.9**	Swimming	78.9	**81.8**	81.7
Parade	33.9	**44.7**	44.0	Rock climbing	**13.8**	14.2	13.7	Biking	67.3	69.0	**70.2**
Parkour	61.3	**72.8**	72.2	Town hall meeting	40.1	36.3	**42.4**	Cat	66.2	68.3	**70.8**
Repairing an appliance	44.5	49.7	**57.4**	Winning a race without a vehicle	22.1	**26.8**	26.7	Dog	65.4	**68.9**	68.6
Working on a sewing project	47.2	**48.0**	46.0	Working on a metal crafts	15.0	17.9	**20.6**	Bird	33.9	35.9	**36.4**
Attempting a bike trick	8.8	**9.3**	8.9	Beekeeping	**54.1**	53.5	48.1	Graduation	25.3	**27.6**	25.1
Cleaning an appliance	10.5	13.2	**15.5**	Wedding shower	19.7	31.5	**40.3**	Birthday	54.2	56.2	**61.9**
Dog show	81.6	84.9	**85.9**	Non-motorized vehicle repair	65.3	66.0	**66.5**	Wedding Reception	17.3	18.3	**19.5**
Giving directions to a location	0.6	**1.0**	0.9	Fixing musical instrument	25.4	31.0	**44.0**	Wedding Ceremony	45.6	53.5	**58.4**
Marriage proposal	0.3	0.4	**0.5**	Horse riding competition	**40.5**	37.9	35.6	Wedding Dance	51.3	58.6	**61.7**
Renovating a home	11.4	12.1	**13.8**	Felling a tree	12.4	16.2	**22.3**	Music Performance	41.0	41.6	**51.1**
Rock climbing	13.7	**14.9**	14.4	Parking a vehicle	17.3	19.5	**21.9**	NonMusic Performance	30.6	35.6	**36.4**
Town hall meeting	40.2	38.9	**44.8**	Playing fetch	1.3	1.3	**1.4**	Parade	49.4	64.0	**64.0**
Winning a race without a vehicle	21.9	**28.8**	27.8	Tailgating	32.9	34.5	**37.7**	Beach	**74.5**	73.9	72.9
Working on a metal crafts	15.2	18.0	**20.4**	Tuning musical instrument	8.7	8.3	**15.1**	Playground	**63.4**	61.9	58.1
mAP	32.4	35.9	**37.1**	*mAP*	24.2	26.0	**28.3**	*mAP*	54.8	58.2	**59.3**

Fusing more modalities to learn Video2vec$^{\mathcal{F}}$ leads to a more effective semantic representation.

5.2 Video2vec Fusion

We evaluate the effectiveness of the Video2vec fusion (Video2vec$^{\mathcal{F}}$) proposed in Section 3.2 for few-example event recognition. We perform the experiments using the appearance, motion, and audio modalities, as detailed in Section 4.2.

5.2.1 Effect of Fusion

We study the impact of fusing multiple modalities for learning the video representation. We start from using only the appearance features (*Appearance*) and gradually add the motion (+ *Motion*) and audio features (+ *Audio*) to learn the video representation by Video2vec$^{\mathcal{F}}$.

Results. Table 5 confirms that incorporating more modalities for estimating the predictability loss leads to a more effective representation learning.

We observe that some events are benefiting more from adding the auditory modality, i.e., *birthday party* and *music performance*. For these events, there are some distinctive terms which are more effectively predicted from the audio features, i.e., `singing`, `clapping`, and `piano`. Other events improve by including motion features, i.e., *parkour* and *parade*, for which some distinctive terms such as `jumping`, `rolling`, and `running` are well predictable from the motion features. It demonstrates that different modalities are complementary for predicting the terms, so they are all required to effectively predict the video descriptions.

5.2.2 Comparison with Other Fusions

We compare Video2vec$^{\mathcal{F}}$ with the following fusion baselines:

1. *Early fusion.* This baseline fuses the modalities by simply concatenating the low-level features from all the three modalities into a longer feature vector. Then the event classifiers are trained and applied on the concatenated low-level features. This simple baseline is shown to be competitive to more complex multiple kernel learning [74].
2. *Late fusion.* This baseline fuses the modalities at the level of event classification scores. A separate event classifier is learned by training an SVM on low-level video features per modality. For each test video, the final event detection score is obtained by averaging the detection scores predicted by each classifier, as evaluated in [4].
3. *Video2vec early fusion.* This baseline learns one Video2vec embedding on the concatenation of various low-level video features, using the standard Video2vec objective of Eq. (1). For each video, a semantic representation is extracted by applying the learned feature projection on the concatenation of the low-level video features. Then, the event classifiers are trained and applied on the semantic video representations.

TABLE 6
Comparison with Other Fusions

	Early fusion	Late fusion	Video2vec early fusion	Video2vec late fusion	Video2vec$^{\mathcal{F}}$
MED 2013	32.6	33.8	33.8	33.6	**37.1**
MED 2014	27.0	27.0	27.0	26.1	**28.3**
CCV$_{10}$	56.3	55.8	56.9	55.4	**59.3**

The Video2vec$^{\mathcal{F}}$ outperforms the alternative fusion tactics.

4. *Video2vec late fusion.* This baseline learns three separate Video2vec embeddings, one per modality, using the standard Video2vec objective of Eq. (1). For each video, a semantic representation is obtained by concatenating the three Video2vec representations, which are predicted from each modality. Then, the event classifiers are trained and applied on the semantic video representations.

Results. The results are reported in Table 6. The Video2vec$^{\mathcal{F}}$ outperforms the alternative fusions on all three test sets. We explain the better performance of Video2vec$^{\mathcal{F}}$ over early fusion and late fusion by the fact that both baselines train the event classifiers directly from low-level video features. However, in Video2vec$^{\mathcal{F}}$ the event classifiers are trained on the semantic Video2vec representations, which are more effective than the low-level features in general, as shown in Section 5.1.3 for the appearance features.

In the Video2vec late fusion the embeddings are learned *separately* per modality. In contrast, Video2vec$^{\mathcal{F}}$ relies on all the video modalities jointly to estimate the predictability loss, which in general is more reliable than the per modality predictability estimations. For further investigation, we visualize the learned textual projection matrices A in Fig. 7. The Video2vec$^{\mathcal{F}}$ prevents some undesirable combination of terms that happen when the predictability losses are estimated separately per modality. For example, in the Video2vec which is learned only on audio features, the terms `laugh`, `cheer`, `bark`, `dog`, and `woof` are all combined as they have similar auditory features (see the right plot). However, the Video2vec$^{\mathcal{F}}$ does not combine the `laugh` and `cheer` with the `bark`, `dog`, and `woof` terms, as these terms are different in the appearance and motion features. Video2vec$^{\mathcal{F}}$ learns a more reliable combination of terms, leading to a more effective video representation.

Video2vec early fusion learns a single feature projection matrix on the concatenated video features. However, the low-level features from different modalities have a different intrinsic dimensionality, distribution, and meaning, which aggravates the learning from their concatenation. In contrast, the Video2vec$^{\mathcal{F}}$ learns separate feature projection matrices per modality, where each feature projection is optimized based on the features from one modality. It alleviates learning the feature projections, which leads to a more effective video representation.

To conclude, the Video2vec$^{\mathcal{F}}$ effectively fuses the features from various video modalities by embedding them into a mutual semantic representation learned jointly over all the modalities.

5.3 Event Specific Video2vec

We evaluate the impact of learning event specific Video2vec embeddings, as proposed in Section 3.3, by comparing the Video2vec$_{\mathcal{E}}$ and its multimodal fusion Video2vec$^{\mathcal{F}}_{\mathcal{E}}$ with *Video2vec* and *Video2vec$^{\mathcal{F}}$*. Different from Video2vec$_{\mathcal{E}}$ and Video2vec$^{\mathcal{F}}_{\mathcal{E}}$, which are learned based on the term sensitive descriptiveness loss, in Eq. (13), the others are learned based on LSI, in Eq. (2).

Results. The results are reported in Tabels 7 and 8. Both event specific embeddings outperform their generic counterparts on both the MED 2013 and MED 2014 test sets. For zero-example event recognition reported in Table 7, the Video2vec$_{\mathcal{E}}$ improves the Video2vec from 15.9 to 18.3 on MED 2013, and from 5.2 to 6.8 on MED 2014. Similarly, the Video2vec$^{\mathcal{F}}_{\mathcal{E}}$ improves the Video2vec$^{\mathcal{F}}$ from 17.8 to 20.0 on MED 2013 and from 6.6 to 8.0 on MED 2014. A consistent improvement is observed for few-example event recognition also albeit with a smaller margin, as detailed in Table 8. The results confirm the effectiveness of learning event specific Video2vec embeddings based on the term sensitive descriptiveness loss.

We explain the lower performance of the Video2vec and Video2vec$^{\mathcal{F}}$ baselines by the fact that the LSI loss treats all the terms equally when measuring the reconstruction error. Hence this loss is biased toward minimizing the reconstruction error for the frequent terms. As a consequence, the Video2vec and Video2vec$^{\mathcal{F}}$ are less accurate for predicting infrequent terms. This drawback is addressed by the term sensitive descriptiveness loss, which minimizes the reconstruction error for the indicative terms, even if those terms are infrequent in the train data. As a result, the Video2vec$_{\mathcal{E}}$ and Video2vec$^{\mathcal{F}}_{\mathcal{E}}$ can predict the distinctive terms of an event more accurately, which leads to an improved event recognition accuracy. In Fig. 8, we compare the term

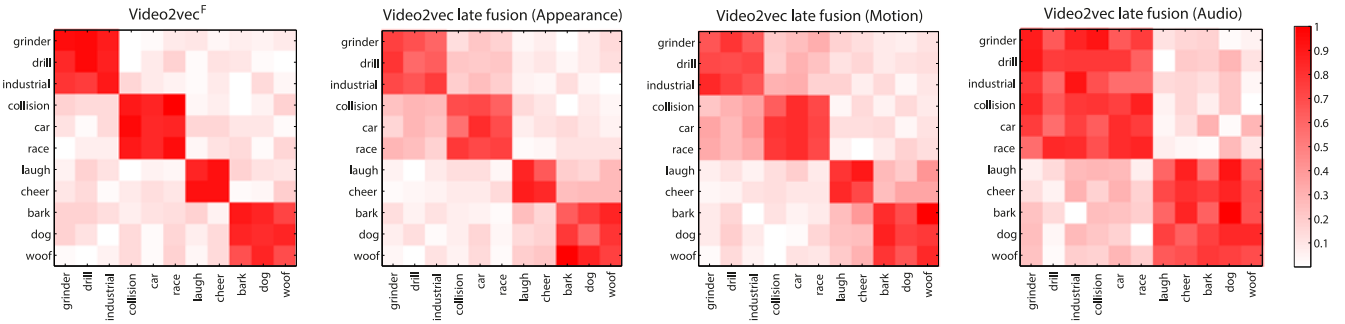

Fig. 7. Effect of learning the embeddings *jointly* over all the modalities by comparing Video2vec$^{\mathcal{F}}$ with the Video2vec late fusion baseline. Each textual projection matrix is visualized by plotting $A \times A^{\top}$, which reveals the learned term combinations. The Video2vec$^{\mathcal{F}}$ learn a more reasonable combination of terms compared to the Video2vec late fusion, where the embeddings are learned disjointedly per modality.

TABLE 7
Effect of Learning Event Specific Video2vec for Zero-Example Event Recognition

MED 2013					MED 2014				
Event	Video2vec	Video2vec$^{\mathcal{F}}$	Video2vec$_{\mathcal{E}}$	Video2vec$^{\mathcal{F}}_{\mathcal{E}}$	*Event*	Video2vec	Video2vec$^{\mathcal{F}}$	Video2vec$_{\mathcal{E}}$	Video2vec$^{\mathcal{F}}_{\mathcal{E}}$
Birthday party	24.6	32.5	30.3	**37.4**	Attempting a bike trick	**8.8**	5.3	7.1	5.9
Changing a vehicle tire	**43.9**	36.0	42.1	38.2	Cleaning an appliance	8.2	10.5	9.5	**12.3**
Flash mob gathering	14.5	30.1	22.4	**33.8**	Dog show	4.0	5.2	**5.8**	5.4
Getting a vehicle unstuck	**40.2**	29.9	36.0	32.2	Giving directions to a location	0.6	0.6	0.6	**1.1**
Grooming an animal	18.7	21.4	**28.0**	23.4	Marriage proposal	0.3	0.7	0.4	**0.7**
Making a sandwich	19.4	15.5	**21.1**	17.1	Renovating a home	5.2	5.1	**6.8**	6.3
Parade	17.6	32.5	26.4	**38.2**	Rock climbing	**1.6**	1.0	1.4	1.0
Parkour	26.1	34.6	28.0	**40.3**	Town hall meeting	1.9	2.3	9.2	**9.2**
Repairing an appliance	39.8	42.4	41.4	**46.3**	Winning a race without a vehicle	**9.4**	7.8	8.4	7.6
Working on a sewing project	30.8	34.3	**36.4**	36.0	Working on a metal crafts	1.5	4.8	4.7	**4.8**
Attempting a bike trick	**8.8**	5.1	7.1	5.9	Beekeeping	0.3	**1.2**	0.6	0.9
Cleaning an appliance	8.2	**12.6**	9.1	12.5	Wedding shower	2.3	2.8	3.3	**3.3**
Dog show	4.0	5.8	5.8	**5.9**	Non-motorized vehicle repair	33.2	39.6	44.2	**46.9**
Giving directions to a location	0.6	0.7	0.7	**0.8**	Fixing musical instrument	4.7	18.3	7.9	**25.5**
Marriage proposal	0.3	**0.8**	0.4	0.7	Horse riding competition	7.7	12.4	12.8	**13.7**
Renovating a home	5.2	5.3	**6.8**	6.4	Felling a tree	7.7	5.7	6.5	**6.8**
Rock climbing	**1.6**	1.1	1.4	1.0	Parking a vehicle	2.8	2.3	2.3	**2.9**
Town hall meeting	1.9	2.5	9.2	**9.6**	Playing fetch	2.2	**3.7**	2.9	3.3
Winning a race without a vehicle	**9.4**	8.1	8.4	7.9	Tailgating	0.2	0.3	0.2	**0.4**
Working on a metal crafts	1.6	4.1	4.9	**7.0**	Tuning musical instrument	0.5	**2.4**	0.9	2.0
mAP	15.9	17.8	18.3	**20.0**	*mAP*	5.2	6.6	6.8	**8.0**

The event specific Video2vec$_{\mathcal{E}}$ and Video2vec$^{\mathcal{F}}_{\mathcal{E}}$ embeddings improve their generic counterparts, Video2vec and Video2vec$^{\mathcal{F}}$. The results demonstrate that the term sensitive descriptiveness loss (L^{ts}_d) and the multimodal predictability loss ($L^{\mathcal{F}}_p$) are both effective and complementary.

vectors, which are predicted by Video2vec and Video2vec$_{\mathcal{E}}$ for two video examples.

We explain the bigger gain for zero-example event recognition by two reasons: First, in few-example event recognition, an SVM event classifier is trained on the provided video exemplars, which makes the recognition somewhat robust to the noise in the video representations. For zero-example recognition the event is recognized by directly matching the event definition with the video representation. Hence, accurate prediction of the terms from videos become crucial. As the second explanation, we highlight that in our experiments the term importance matrix $\boldsymbol{H}_e$ is determined based on the event definitions. This is reasonable for zero-example event recognition, where the events are recognized using their definition only. However, for few-example event recognition it might be more effective to learn the term importance matrix $\boldsymbol{H}_e$ from the video examples rather than pre-define them from the event definitions.

TABLE 8
Effect of Learning Event Specific Video2vec for Few-Example Event Recognition

	Video2vec	Video2vec$^{\mathcal{F}}$	Video2vec$_{\mathcal{E}}$	Video2vec$^{\mathcal{F}}_{\mathcal{E}}$
MED 2013	32.4	37.1	32.6	**37.8**
MED 2014	24.2	28.3	24.3	**29.1**

Confirming the conclusion of Table 7.

Finally, the most effective representation is learned by the Video2vec$^{\mathcal{F}}_{\mathcal{E}}$. It indicates that the term sensitive descriptiveness loss (L^{ts}_d) and the multimodal predictability loss ($L^{\mathcal{F}}_p$) are both effective and complementary when learning semantic video representations for event recognition.

5.4 Comparison with the State-of-the-Art

We evaluate the merit of Video2vec by comparing it with several other recent works on few- and zero-example event recognition. Since in most papers the results are reported only on the MED 2013 test set, we limit our comparisons to this test set. Each approach may vary the data sources used for representation pre-training, yet they all abide to the standard data partitioning for training and testing of event classifiers as prescribed by the NIST evaluation protocol.

The results are reported in Table 9. Our proposed representation learning sets a new state-of-the-art for the both few- and zero-example event recognition. It should be noted that these works rely on modeling the temporal aspects of the events [80], [81], [82], [83], using larger train set for representation learning [29], [62], [66], and query expansion [31], [66] to improve the event recognition. These improvements can also be applied together with our representation learning. Very recently Jiang et al. [29] improved their results for zero-example recognition from 18.3 to 20.8 after adding re-ranking by pseudo relevance feedback, we expect a similar gain for Video2vec$^{\mathcal{F}}_{\mathcal{E}}$.

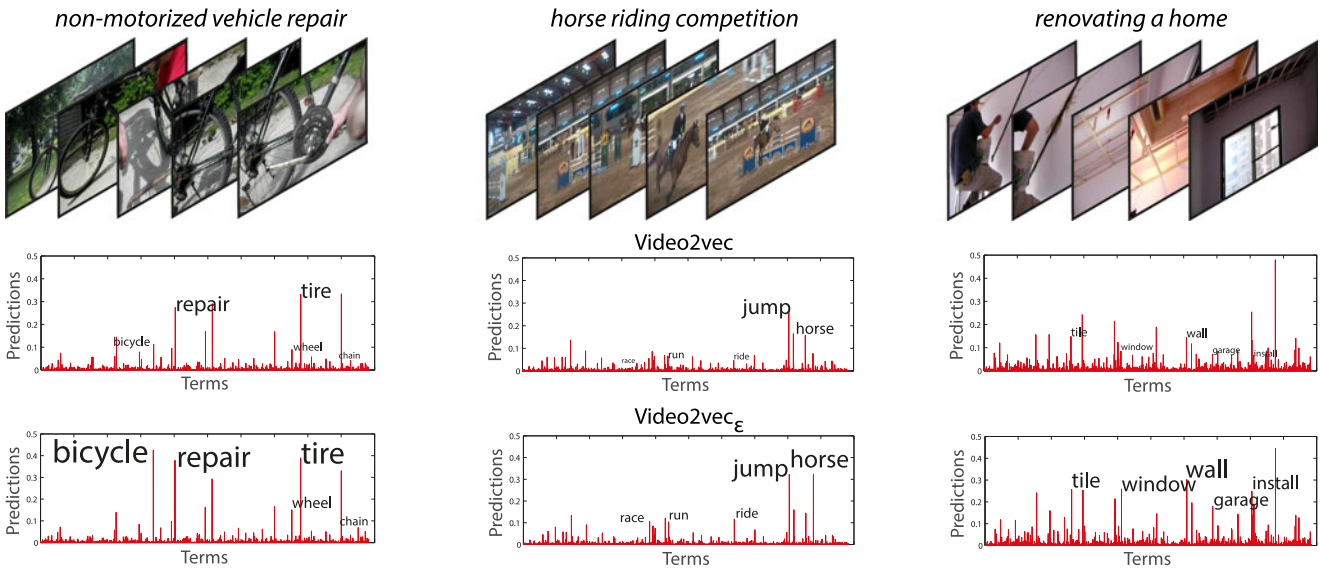

Fig. 8. Unseen video examples and their term vectors predicted by the Video2vec (middle) and the Video2vec$_\mathcal{E}$ embeddings (bottom). The size of each term indicates its prediction confidence score. The Video2vec$_\mathcal{E}$ more effectively predicts the indicative terms of the event.

We also perform zero-example event recognition using only the event name as query, rather than the full definition, following the setting of [84], [85]. The results are reported in Table 10. Surprisingly, by using just event names as query, the Video2vec$^{\mathcal{F}}_{\mathcal{E}}$ performance is increased from 20.0 to 22.5 mAP. The improvement is most significant for the events with informative names, i.e., *flash mob gathering* and *parkour*, whose AP increases from 33.8 and 40.3 to 46.8 and 64.2 respectively. For some events such as *repairing an appliance* and *cleaning an appliance*, the event definitions contain specific informative words, i.e., `microwave`, `stove` and `oven`, which are crucial for recognizing the event. Hence, the event definitions are more effective for recognizing these events. Finally, Video2vec$^{\mathcal{F}}_{\mathcal{E}}$ is more effective than [84] and [85] for zero-example recognition using event names.

TABLE 9
Comparison with the State-of-the-Art

Few-example on MED 2013		Zero-example on MED 2013	
Habibian et al. [37]	19.6	Ye et al. [34]	9.0
Nagel et al. [43]	21.8	Chang et al. [66]	9.6
Li et al. [80]†	23.7	Mazloom et al. [64]	11.9
Tang et al. [81]†	26.8	Wu et al. [31]	12.7
Sun et al. [82]†	28.7	Jiang et al. [62]	12.9
Chang et al. [83]	30.9	Jiang et al. [29]	18.3
This paper: *Video2vec*$^{\mathcal{F}}_{\mathcal{E}}$	**37.8**	This paper: *Video2vec*$^{\mathcal{F}}_{\mathcal{E}}$	**20.0**

† *Based on implementation by [83]*
Video2vec embeddings set a new state-of-the-art for the both few- and zero-example event recognition.

TABLE 10
Zero-Example Event Recognition on MED 2013
by Querying with Event Names Instead of Event Definitions

	Event Definitions	Event Names
Jain et al. [84]	NA	4.2
Han et al. [85]	NA	16.4
This paper: *Video2vec*$^{\mathcal{F}}_{\mathcal{E}}$	20.0	**22.5**

6 Conclusion

In this paper, we attack the problem of event recognition in video when examples are scarce. We propose the *Video2vec* embedding that learns a semantic video representation from a set of videos and their textual descriptions by minimizing a joint objective function balancing term descriptiveness and video predictability losses. As a result, the words which are correlated in the descriptions are combined together to improve their video predictability.

In addition, we propose the *Video2vec*$^{\mathcal{F}}$ embedding with a multimodal predictability loss learned jointly over video appearance, motion, and audio features. The different modalities are complementary for predicting the words, so they are all required for a richer video description. Moreover, embedding the heterogeneous video features into a mutual semantic space leads to few-example event recognition that is more effective than traditional fusion tactics.

We also propose the *Video2vec*$_{\mathcal{E}}$ embedding to learn event specific video representations. This embedding relies on a term sensitive descriptiveness loss to learn a more accurate representation for the indicative words.

Finally, the *Video2vec*$^{\mathcal{F}}_{\mathcal{E}}$ embedding demonstrates that the term sensitive descriptiveness loss and the multimodal predictability loss are both effective and complementary to learn semantic video representations for few- and zero-example event recognition with state-of-the-art accuracy.

We consider *Video2vec*'s ability to generate human interpretable representations for previously unseen videos most appealing, as it opens up new connections with natural language processing and computational linguistics for describing and querying videos.

References

[1] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative CNN video representation for event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1798–1807.
[2] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *Int. J. Comput. Vis.*, vol. 119, pp. 219—238, 2015.

[3] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, "High-level event recognition in unconstrained videos," *Int. J. Multimedia Inf. Retrieval*, vol. 2, pp. 73–101, 2013.

[4] P. Natarajan, et al., "Multimodal feature fusion for robust event detection in Web videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1298–1305.

[5] A. Tamrakar, et al., "Evaluation of low-level features and their combinations for complex event detection in open source videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3681–3688.

[6] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 140–153.

[7] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[8] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1725–1732.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[10] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[11] P. Over, et al., "TRECVID 2013-an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proc. TRECVID Workshop*, 2013.

[12] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 3551–3558.

[13] P. Natarajan, et al., "BBN VISER TRECVID 2013 multimedia event detection and multimedia event recounting systems," in *Proc. TRECVID Workshop*, 2013.

[14] S. Oh, et al., "Multimedia event detection with multimodal feature fusion and temporal concept localization," *Mach. Vis. Appl.*, vol. 25, pp. 49–69, 2014.

[15] S.-I. Yu, et al., "Informedia@ TRECVID 2014 MED and MER," in *Proc. TRECVID Workshop*, 2014.

[16] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Proc. 22nd Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1410–1418.

[17] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 951–958.

[18] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1778–1785.

[19] M. Merler, B. Huang, L. Xie, G. Hua, and A. Natsev, "Semantic model vectors for complex video event recognition," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 88–101, Feb. 2012.

[20] H. Izadinia and M. Shah, "Recognizing complex events using large margin joint low-level event model," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 430–444.

[21] Z. Ma, "From concepts to events: A progressive process for multimedia content analysis," Ph.D. dissertation, Dept. Inf. Eng. Comput. Sci., Univ. Trento, Trento, Italy, 2013.

[22] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. Hauptmann, "Complex event detection via multi-source video attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2627–2633.

[23] A. Habibian and C. G. M. Snoek, "Recommendations for recognizing video events by concept vocabularies," *Comput. Vis. Image Understanding*, vol. 124, pp. 110–122, 2014.

[24] M. Mazloom, E. Gavves, and C. G. M. Snoek, "Conceptlets: Selective semantics for classifying video events," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2214–2228, Dec. 2014.

[25] Z. Ma, Y. Yang, N. Sebe, and A. G. Hauptmann, "Knowledge adaptation with partially shared features for event detection using few exemplars," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1789–1802, Sep. 2014.

[26] S. Bhattacharya, M. M. Kalayeh, R. Sukthankar, and M. Shah, "Recognition of complex events: Exploiting temporal dynamics between underlying concepts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2243–2250.

[27] A. Dehghan, H. Idrees, and M. Shah, "Improving semantic concept detection through the dictionary of visually-distinct elements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2585–2592.

[28] Y. Yan, et al., "Complex event detection via event oriented dictionary learning," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 3841–3847.

[29] 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 *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 49–58.

[30] O. Russakovsky, et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.

[31] S. Wu, S. Bondugula, F. Luisier, X. Zhuang, and P. Natarajan, "Zero-shot event detection using multi-modal fusion of weakly supervised concepts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2665–2672.

[32] Y. Cui, D. Liu, J. Chen, and S.-F. Chang, "Building a large concept bank for representing events in video," *arXiv:1403.7591*, 2014.

[33] J. Chen, Y. Cui, G. Ye, D. Liu, and S.-F. Chang, "Event-driven semantic concept discovery by exploiting weakly tagged internet images," in *Proc. Int. Conf. Multimedia Retrieval*, 2014, pp. 1:1–1:8.

[34] 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 *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 471–480.

[35] T. Berg, A. Berg, and J. Shih, "Automatic attribute discovery and characterization from noisy Web data," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 663–676.

[36] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[37] A. Habibian, T. Mensink, and C. G. M. Snoek, "VideoStory: A new multimedia embedding for few-example recognition and translation of events," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 17–26.

[38] C. Szegedy, et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[39] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[40] D. Oneata, J. Verbeek, and C. Schmid, "Action and event recognition with Fisher vectors on a compact feature set," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1817–1824.

[41] Y.-G. Jiang, et al., "Columbia-UCF TRECVID 2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching," in *Proc. TRECVID Workshop*, 2010.

[42] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[43] M. Nagel, T. Mensink, and C. G. M. Snoek, "Event Fisher vectors: Robust encoding visual diversity of visual streams," in *Proc. British Mach. Vis. Conf.*, 2015, pp. 178.1–178.12.

[44] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the Fisher vector: Theory and practice," *Int. J. Comput. Vis.*, vol. 105, pp. 222–245, 2013.

[45] V. Ferrari and A. Zisserman, "Learning visual attributes," in *Proc. Advances Neural Inf. Process. Syst.*, 2007, pp. 433–440.

[46] D. Blei and M. Jordan, "Modeling annotated data," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 127–134.

[47] J. Weston, S. Bengio, and N. Usunier, "WSABIE: Scaling up to large vocabulary image annotation," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 2764–2770.

[48] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for attribute-based classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 819–826.

[49] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3128–3137.

[50] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Language Technol.*, 2015, pp. 1494–1504.

[51] P. Das, C. Xu, R. Doell, and J. Corso, "A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2634–2641.

[52] R. Xu, C. Xiong, W. Chen, and J. J. Corso, "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2346–2352.
[53] J. Costa Pereira, et al., "On the role of correlation and abstraction in cross-modal multimedia retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 521–535, Mar. 2014.
[54] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 689–696.
[55] F. Feng, X. Wang, and R. Li, "Cross-modal retrieval with correspondence autoencoder," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 7–16.
[56] H. Hotelling, "Relation between two sets of variates," *Biometrika*, vol. 28, pp. 322–377, 1936.
[57] D. Putthividhy, H. T. Attias, and S. S. Nagarajan, "Topic regression multi-modal latent Dirichlet allocation for image annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3408–3415.
[58] P. Das, R. Srihari, and J. Corso, "Translating related words to videos and back through latent topics," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 485–494.
[59] Y. Aytar, M. Shah, and J. Luo, "Utilizing semantic word similarity measures for video retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[60] D. Parikh and K. Grauman, "Relative attributes," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 503–510.
[61] L. Jiang, T. Mitamura, S.-I. Yu, and A. G. Hauptmann, "Zero-example event search using multimodal pseudo relevance feedback," in *Proc. Int. Conf. Multimedia Retrieval*, 2014, Art. no. 297.
[62] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2694–2700.
[63] A. Habibian, T. Mensink, and C. G. M. Snoek, "Composite concept discovery for zero-shot video event detection," in *Proc. Int. Conf. Multimedia Retrieval*, 2014, Art. no. 17.
[64] M. Mazloom, A. Habibian, D. Liu, C. G. M. Snoek, and S.-F. Chang, "Encoding concept prototypes for video event detection and summarization," in *Proc. 5th ACM Int. Conf. Multimedia Retrieval*, 2015, pp. 123–130.
[65] J. Dalton, J. Allan, and P. Mirajkar, "Zero-shot video retrieval using content and concepts," in *Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 1857–1860.
[66] X. Chang, Y. Yang, A. G. Hauptmann, E. P. Xing, and Y.-L. Yu, "Semantic concept discovery for large-scale zero-shot event detection," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2234–2240.
[67] Q. Wang, J. Xu, H. Li, and N. Craswell, "Regularized latent semantic indexing," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 685–694.
[68] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1999, pp. 50–57.
[69] J. Mairal, F. Bach, and J. Ponce, "Sparse modeling for image and vision processing," *Found. Trends Comput. Graph. Vis.*, vol. 8, no. 2/3, pp. 85–283, 2012.
[70] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables,"*J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, 2005.
[71] K. Q. Weinberger and O. Chapelle, "Large margin taxonomy embedding for document categorization," in *Proc. 21st Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1737–1744.
[72] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.*, 2010, pp. 177–186.
[73] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 399–402.
[74] K. Tang, B. Yao, L. Fei-Fei, and D. Koller, "Combining the right features for complex event recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2696–2703.
[75] S. Strassel, et al., "Creating HAVIC: Heterogeneous audio visual internet collection," in *Proc. 8th Int. Conf. Language Resources Eval.*, 2012, pp. 2573–2577.
[76] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. Loui, "Consumer video understanding: A benchmark database and an evaluation of human and machine performance," in *Proc. 1st ACM Int. Conf. Multimedia Retrieval*, 2011, Art. no. 29.
[77] Y. Jia, et al., "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[78] M. Jain, H. Jégou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2555–2562.
[79] CMU sphinx open source toolkit for speech recognition. [Online]. Available: http://cmusphinx.sourceforge.net/, Accessed on: Aug. 3, 2015.
[80] W. Li, Q. Yu, A. Divakaran, and N. Vasconcelos, "Dynamic pooling for complex event recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2728–2735.
[81] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1250–1257.
[82] C. Sun and R. Nevatia, "DISCOVER: Discovering important segments for classification of video events and recounting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2569–2576.
[83] X. Chang, Y.-L. Yu, Y. Yang, and A. G. Hauptmann, "Searching persuasively: Joint event detection and evidence recounting with limited supervision," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 581–590.
[84] M. Jain, J. C. van Gemert, T. Mensink, and C. G. M. Snoek, "Objects2action: Classifying and localizing actions without any video example," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4588–4596.
[85] X. Han, B. Singh, V. I. Morariu, and L. S. Davis, "VRFP: On-the-fly video retrieval using web images and fast fisher vector products," *arXiv:1512.03384*, 2016.

Amirhossein Habibian received the BSc degree in computer engineering and the MSc degree in artificial intelligence from the University of Tehran, Iran, in 2008 and 2011, respectively. Currently, he is working toward the PhD degree at the University of Amsterdam, The Netherlands. He is also a senior engineer with Qualcomm Research, Amsterdam, working on deep learning and computer vision. He received the Best Paper Award of ACM Multimedia 2014. His research interests include multimedia retrieval, computer vision, and machine learning.

Thomas Mensink received the MSc (cum laude) degree in artificial intelligence from the University of Amsterdam, The Netherlands, in 2007 and the PhD degree in computer science from the University of Grenoble, France, in 2012, working with the LEAR team of INRIA Grenoble and Xerox Research Centre Europe. Since 2012, he is a postdoc researcher with the University of Amsterdam. In 2015, he received a prestigious NWO VENI grant. His research interests include machine learning and computer vision.

Cees G. M. Snoek received the MSc degree in business information systems and the PhD degree in computer science from the University of Amsterdam, The Netherlands, in 2000 and 2005, respectively. He is currently a director of the QUVA Lab, Research Lab of Qualcomm, and University of Amsterdam, on deep learning and computer vision. He is also a principal engineer with Qualcomm and an associate professor. His research interests focus on video and image recognition. He received the NWO Veni award (2008), a Fulbright Junior Scholarship (2010), an NWO Vidi award (2012), and the Netherlands Prize for ICT Research (2012). He is a senior member of the IEEE.