A Fine Granularity Object-Level Representation for Event Detection and Recounting

Hao Zhang and Chong-Wah Ngo, *Member, IEEE*

***Abstract*—Multimedia events such as "birthday party" usually involve the complex interaction between humans and objects. Unlike actions and sports, these events rarely contain unique motion patterns to be vividly explored for recognition. To encode rich objects in the events, a common practice is to tag an individual video frame with object labels, represented as a vector signifying probabilities of object appearances. These vectors are then pooled across frames to obtain a video-level representation. The current practices suffer from two deficiencies due to the direct employment of deep convolutional neural network (DCNN) and standard feature pooling techniques. First, the use of max-pooling and softmax layers in DCNN overemphasize the primary object or scene in a frame, producing a sparse vector that overlooks the existence of secondary or small-size objects. Second, feature pooling by max or average operator over sparse vectors makes the video-level feature unpredictable in modeling the object composition of an event. To address these problems, this paper proposes a new video representation, named Object-VLAD, which treats each object equally and encodes them into a vector for multimedia event detection. Furthermore, the vector can be flexibly decoded to identify evidences such as key objects to recount the reason why a video is retrieved for an event of interest. Experiments conducted on MED13 and MED14 datasets verify the merit of Object-VLAD by consistently outperforming several state-of-the-arts in both event detection and recounting.**

***Index Terms*—Multimedia event detection and recounting, object encoding, search result reasoning.**

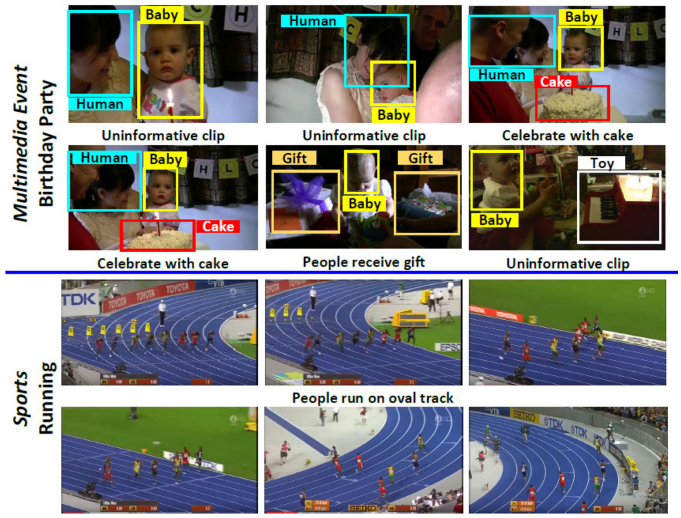


Fig. 1. Comparison of multimedia event and sport activities. Event "birthday party" consists of a sequence of interactions: "people sing birthday song", "celebrate with cake" and "people receive gift". Each interaction contains evidential objects (red and dark yellow bounding boxes). Sport "running" only consists one interaction "people run on oval track". The figure is best viewed in color.

I. Introduction

SEMANTIC understanding of video content is traditionally a challenging problem in multimedia computing [1]–[4], especially with the massive growth of user-generated videos where there are no constraints on content. In the past few years, significant progress has been made for retrieval of action and sport videos by motion features (e.g., motion-relativity [5], improved dense trajectory [6] and 3D convolutional networks [7]). In standard benchmark datasets such as Sport-1M [8] and UCF-101 [9], these videos share the characteristics of distinctive motion patterns in short duration. Extending these works for user-generated videos which are rich of multimedia events, nevertheless, is not straightforward. Multimedia event, in contrast, is complex in motion patterns and object composition, not mentioning visual diversities in event content that can span a large temporal scale [10]. Fig. 1 shows video examples of "birthday party" which are rich of activities like "singing birthday song" and objects like "cake", "candle" and "toy". Compared to sport action like "running", higher-level feature representation beyond motion is generally expected to depict the rich and diverse event content.

This paper addresses multimedia event detection (MED) and event recounting (MER). MED retrieves videos containing a target event, while MER explains the reasons why videos are retrieved by prompting thumbnails relevant to the event. The provided thumbnails ideally help users to quickly locate the positive videos from the imperfect search result. The past efforts are mostly devoted to the derivation of effective features encapsulating the rich semantic content in videos [11]–[14]. Particularly, with the advancement in deep learning, impressive performance has been reported by leveraging features in the deep convolutional neural networks (DCNN) [15]–[17] for event classifier training. These features include the semantic labels extracted from the output layer of DCNN [12], [13] and the spatial information from convolutional layers [14]. In [12], [13],

Manuscript received December 13, 2017; revised July 15, 2018; accepted October 29, 2018. Date of publication December 3, 2018; date of current version May 22, 2019. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Grant CityU 120213. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xilin Chen. *(Corresponding author: Hao Zhang.)*

The authors are with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: hzhang57-c@my.cityu.edu.hk; cwngo@cs.cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2884478

the DCNN predicted scores from as large as 15,000 object categories are encoded as video descriptor. The empirical studies confirm that object information matters in event [12] and action [13] recognition and complements with motion feature. In [14], the descriptors extracted from feature maps of DCNN are encoded with VLAD (Vector of Linearly Aggregated Descriptors) [18] and impressive performance is reported on TRECVID MED/MER benchmarks [10].

Despite encouraging performance, an issue often overlooked by the existing approaches is that DCNN is trained with single-label classification [15]–[17]. Specifically, due to the use of max pooling and softmax layers, the network is learned to recognize one target label for an input. As a sequence, the neural activations are tuned to highlight the primary object or scene while masking out secondary object information. This results in a very sparse prediction of scores especially when the output layer contains a large number of semantic labels. In other words, the score distribution is ineffective in reflecting object composition. Directly employing the DCNN semantic labels as in [12], [13] is helpful for action recognition, but can potentially limit MED performance due to loss of secondary object information. We refer this problem as "loss-in-object-details". Utilizing mid-level features as in [14] bypasses the side-effect of softmax layer, but still, the activation of small-size objects will be overwhelmed by max-pooling operation in DCNN.

This paper addresses the problem of "loss in object details" in DCNN, by proposing a new feature Object-VLAD to describe object composition in videos. During encoding stage, Object-VLAD samples frames and object candidates, representing the primary and secondary object information in a video. Each of them is first described with DCNN mid-layer feature, and then encoded with VLAD into a compact video descriptor for MED. During decoding stage, the descriptors of retrieved videos are unrolled to locate frames and objects as evidences for MER. The main contribution of this paper is the proposal of Object-VLAD, a simple yet effective feature generated using off-the-shelf techniques, in alleviating the problem of using DCNN features for multimedia events which are rich of object in different scales. We empirically verify the power of Object-VLAD in preserving secondary object information through MED datasets [10]. On both MED and MER, Object-VLAD shows preferable performances compared to several existing techniques.

The remaining of this paper is organized as below. Section II presents the related work in MED and MER. Section III motivates the study in this paper by empirically quantifying the loss-in-details in DCNN network. Section IV describes the generation of Object-VLAD in encoding primary and secondary object information. Section V elaborates the use of Object-VLAD for MED and MER. Sections VI and VII present experimental setup and justify the effectiveness of Object-VLAD. Finally, Section VIII concludes this paper.

II. Related Works

The challenge of MED comes from capturing the rich but diverse visual content in videos. Two essential steps include the extraction of event-relevant features and encoding of the features for event classifier learning. Along this direction, various studies include feature extraction [5], [6], [19]–[22], discriminative learning [23]–[25] and multi-modality modeling [26]–[28] have been explored. The most related research efforts to this paper are feature coding [14] and object pooling [29].

Early efforts in feature extraction include the sampling of salient points described with SIFT descriptors from video keyframes, followed by the encoding of the descriptors as bag-of-visual-words (BoW) with spatial pyramid representation [19], [20]. In addition to static visual patterns, videos naturally contain motions across multiple frames, in [5], Wang *et al.* propose motion-relativity that accumulates relative movement between visual words to capture dynamic information in user-generated videos for event retrieval. Improved dense trajectory (IDT) encoded with Fisher vector (FV) is later introduced to capture the video dynamics and shows promising performance for action and event detection [6]. However, the extraction of these features, especially IDT, is computationally expensive. DCNN features, which are efficient to extract from fully connected and convolutional layers, exhibit strong performance for multimedia event detection as shown in [14], [29], [30].

Subsequently, due to the development of large concept banks such as Concepts-280 [31], SIN-346 [32], ImageNet [15], ImageNet-Shuffle [12] and EventNet [21], video content is described by concept distribution, encoded as a semantical vector of which each dimension corresponds to the prediction score of a concept classifier. Compared to low-level features such as SIFT and IDT, each vector dimension carries a semantic meaning and hence event detection is feasible even when there is very few or no training examples [33], [34]. Variants of approaches have been proposed for optimization of this semantic representation. Examples include conceptlet [35] which mines a small subset of event-relevant concepts for event classifier learning, and concept prototypes [22] which consider the visual diversity of a concept in generating a semantical vector.

In practice, only a portion of video fragments is discriminative for event description. Instead of using the entire video, discriminative learning aims to exclude irrelevant or noise portions from classifier learning. For example, in [23], classifier is learnt to simultaneously identify event-relevant fragments for robust model training. In [24], linear discriminant analysis (LDA) is employed for mining discriminative fragments from training videos. The fragments are leveraged for learning codebook to encode videos into bag-of-fragments (BoF) for classifier learning. Instead of eliminating indiscriminative fragments as [24], the approach in [25] weights the importance of fragments for classifier learning. The importance is determined by concepts which are deemed to be relevant to the textual event description by word2vec measure.

Multi-modal approaches leverage text and audio for event detection. In [26], Zhang *et al.* adopt FrameNet, which is a semantical resource defining daily event scenes, to select relevant visual concepts from textual event description for capturing video contents. Since concept detectors are trained on web images and then applied on videos, domain adaption technique is used in the concept detector training phase to reduce the

gap between image and video domain. VideoStory [27] learns the projections of hand-crafted motion feature and textual description into a cross-modal joint space for feature embedding. Ideally, the joint space clusters motions into topics and hence the embedded feature is more powerful in capturing event context. And, [28] exploits correlations between patches randomly selected from audio signals and video frames by auto-encoder for event detection.

Our work is closely related to CNN-VLAD [14], and object pooling [29]. CNN-VLAD applies spatial pyramid pooling on the feature map of DCNN to preserve spatial information. The resulting feature is encoded with VLAD and impressive performance is reported. NetVLAD [36], instead of offline learning visual codebook as CNN-VLAD, incorporates a novel layer on top of DCNN for end-to-end learning of codebook. However, NetVLAD requires sufficient training examples for performance guarantee. As reported in [37], NetVLAD is outperformed by CNN-VLAD on TRECVID MED benchmark dataset due to lack of sufficient positive training examples. Object pooling [29], which addresses the problem of object composition as this paper, max pools the probability or fully connected features of DCNN extracted from candidate object regions of a video. Different from [29], our work exploits mid-level features encoded with VLAD for classifier learning.

Other research efforts include temporal modeling [38], weakly supervised learning [11], [39], hard example mining [40], and improvement of speed efficiency [41]. In [38], MUT-based recurrent network is employed to model the temporal dynamics of videos for detection. Weakly supervised learning is studied in [11], [39] by harnessing web images and videos. In [11], given a textual event query, Han *et al.* utilize a commercial search engine to propose relevant web images, then rank candidate videos according to their visual similarity to these images. Both images and videos are first described by DCNN feature and then encoded by the Fisher vector for generating fixed-length feature representation. To address the problem of domain shift in learning, Gan et al. propose a lead-and-exceed network to prune noisy web images and videos [39]. In [40], Ma et al. observe that negative examples resemble positive videos in different degrees, and hence propose to iteratively assign soft labels for hard negative examples to benefit classifier learning. Jiang *et al.* in [41] provide a comprehensive study on how features, classifiers, and fusion strategies affect computational efficiency in the current multimodal event detection baseline, and also study the minimum number of visual/audio frames required to ensure a good detection performance.

The existing approaches in MER generate either text [42]–[45] or visual [24], [29], [46]–[50] snippets as evidences. Except for [42] which synthesizes sentences by knowledge ontology, the former approaches [43]–[45] mostly prompt a sparse set of concepts detected in testing videos as evidence. The concepts are pre-determined as relevant and discriminative during training from event description or positive samples. The latter approaches prompt keyframes as thumbnails or short video fragments as summaries to recount events. A common way is to locate the fragments [24], [46]–[48], [50] or objects [29] that exhibit high responses to event-relevant concepts. More sophisticated approach such as [48] defines local evidence templates by fragment clustering and formulates recounting as a quadratic programming problem. DevNet [46] backprojects the predicted score of an event into the feature map of DCNN and locates pixels that generate high activations as spatial evidence. Similar in spirit, heat map is superimposed on evidential frames to indicate regions relevant to an event [49]. In addition, [50] considers the relevancy and diversity of evidences, and formulates the selection of evidences as an integer linear programming problem.

Compared to the existing techniques, the novelty of Object-VLAD lies in integrating the strengths of different techniques. Like [24], [44], [47], encoding and decoding of features is allowed, but Object-VLAD is a more powerful feature for capturing object information. Although using the same mid-level features and encoding scheme as CNN-VLAD [14], Object-VLAD encodes object composition in such a way that secondary objects overlooked by DCNN can still be captured. This issue is not addressed in CNN-VLAD for encoding only frame-level feature map of DCNN, where the neural responses of secondary objects can be overwhelmed by primary object. Finally, different from our prior work [29] which accumulates object-level information with max pooling, Object-VLAD utilizes a more sophisticated pooling technique based on VLAD encoding. Object-VLAD represents the first study that incorporates object proposal, deep feature and VLAD encoding for seamless encoding of object evidences. Furthermore, our work also results in a new VLAD decoding algorithm that can effectively enumerate the contributions of objects to a query event for recounting.

III. LOSS IN DETAILS

MED requires a fine-granularity understanding of object cluster in space and time to make sense of an event. The existing DCNN, nevertheless, is deliberately learned in such a way that primary object, which occupies a larger portion of the scene, can be emerged out of surrounding smaller-size objects for supervised classification. This is essentially due to the use of max pooling and softmax layer in enforcing the response from primary object. While beneficial for single-label classification, the learning procedure has overwhelmed the activation of non-dominant objects. Direct use of DCNN features will result in over-emphasis of large-size objects and limit the descriptiveness for MED and MER.

We design an experiment using deep architecture pre-trained on ImageNet-1000 [15], as shown in Fig. 2, to quantify the "loss-in-details". The experiment estimates the amount of regional information being filtered out, by passing video frames with two different strategies through the network. Initially, RoIs (regions of interest) of frames are located. The first strategy extracts and scales each RoI to the original frame size, followed by feeding to the network to generate the "feature map of scaled RoI", denoted as f_1. The second strategy, instead, inputs the video frame to generate a feature map. The RoIs are then extracted from the feature map and scaled accordingly to obtain the "scaled feature map of RoI", denoted as f_2. We adopt the same strategy used in [51], [52] to scale up each sub-feature

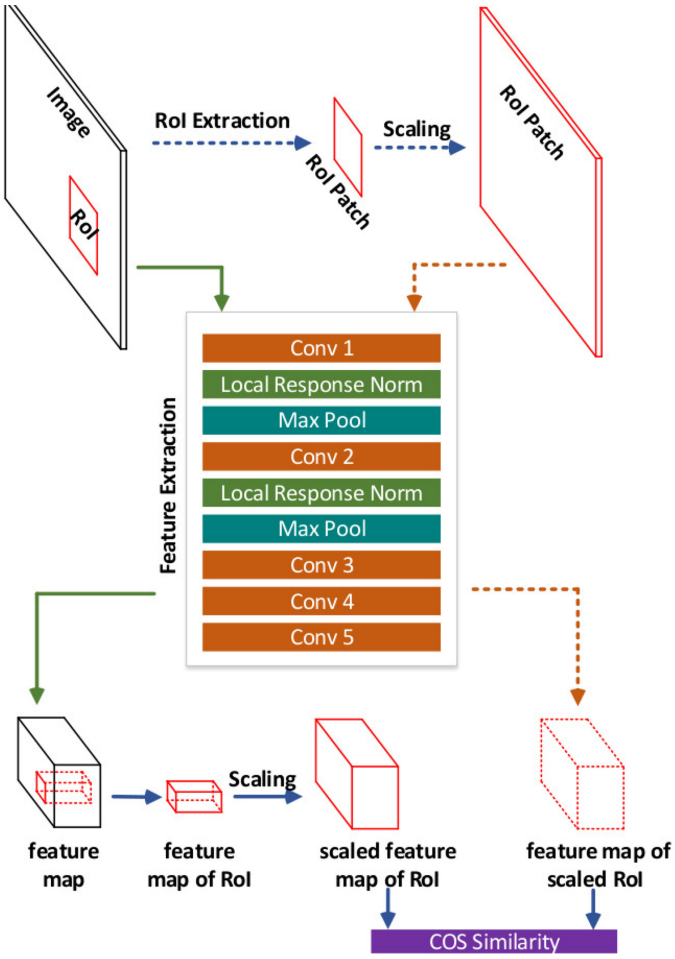


Fig. 2. An experiment designed for quantifying the loss of regional information on convolutional neural network. Two different inputs are compared: video frame (solid line) and RoI extracted from the frame (dotted line). The difference between the two inputs is estimated through "scaled feature map of RoI" and "feature map of scaled RoI". The figure is best viewed in color.

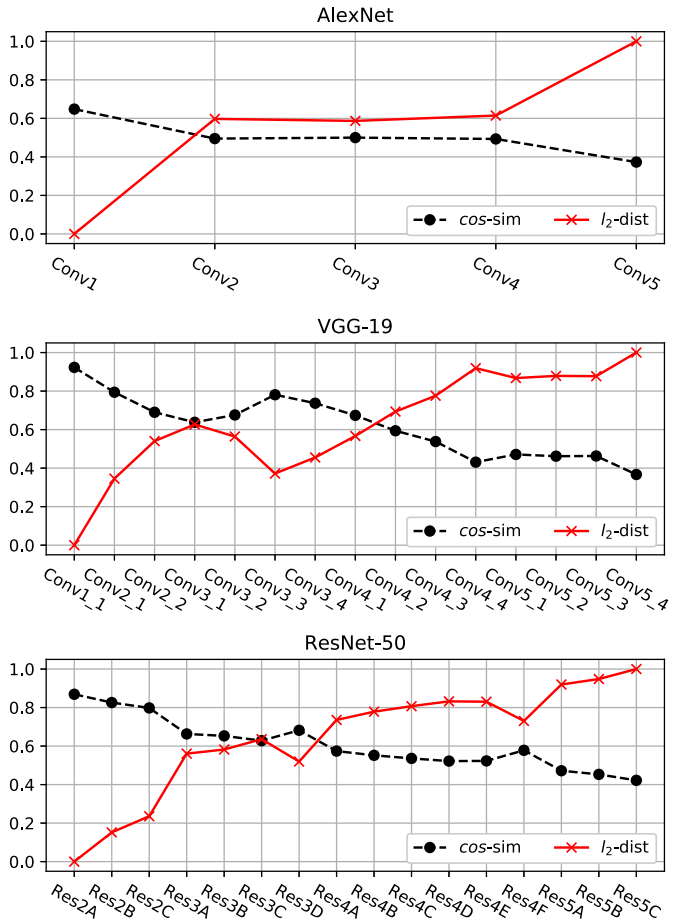


Fig. 3. Euclidean distance (i.e., l_2) and cosine similarity between "scaled feature map of RoI" and "feature map of scaled RoI" with different types of features. Three deep architectures (AlexNet, VGG-19-Net, Deep ResNet-50) are used.

map of RoI. We expect that the neural activation of f_2 should be similar to f_1, where background context is removed before feeding to DCNN, across different layers of network. In other words, information loss is quantized by the dissimilarity between f_1 and f_2, which provides cue on the degree in which the neural response of f_2 is impacted by surrounding background of RoI. To this end, we estimate the information loss between f_1 and f_2 by comparing their cosine similarity and Euclidean distance. The Euclidean distances are min-max normalized into the 0–1 range. Both metrics are popularly used for assessing the similarity or distance between DCNN features for image retrieval [36], [53] and patch matching [54].

The experiment is conducted using 8,852 RoIs from 415 frames of 10 videos on three separate networks, AlexNet [15], VGG-19 [16] and ResNet-50 [17]. Fig. 3 shows the distances and similarities of RoI feature maps extracted at different layers of the network. The distances and similarities shown are averaged over all the extracted RoIs. The general trend in these networks is that dissimilarity and distance keep increasing, implying the loss of regional information in f_2, as going deeper into the network. Particularly, a sharper drop of similarity (or local minimum) is observed near max-pooling layers, for example, conv2-2 and conv3-1 in VGG-19, and Res2C and Res3A in ResNet-50. Based on the similarity drop that reaches the value of around 0.4 for all three networks, we estimate that 60% of regional activation is overwhelmed in f_2 by primary objects. The experiment basically verifies our claim in information loss due to the use of single-label DCNN, and the loss is estimated to be as high as 60%.

Figure 4 further visualizes f_1 and f_2 as heat maps, which are generated by average pooling of neural responses across different channels of a feature map. As shown in the middle column, the primary object receives higher neural activation than secondary RoI, resulting in low response of f_2. In contrast, f_1 (right), which is cropped and input to DCNN, still preserves high neural response for secondary RoI.

While the deep network can plausibly be remedied with loss function tailored for multi-label classification, there is no off-the-shelf network available yet for classification up to a scale of 1,000 multiple labels. Manually collecting a large number of positive examples for networking training is extremely time-consuming and cost expensive. Instead, we propose a new feature, named Object-VLAD, that aggregates regional information for depicting the rich object composition in multimedia events.

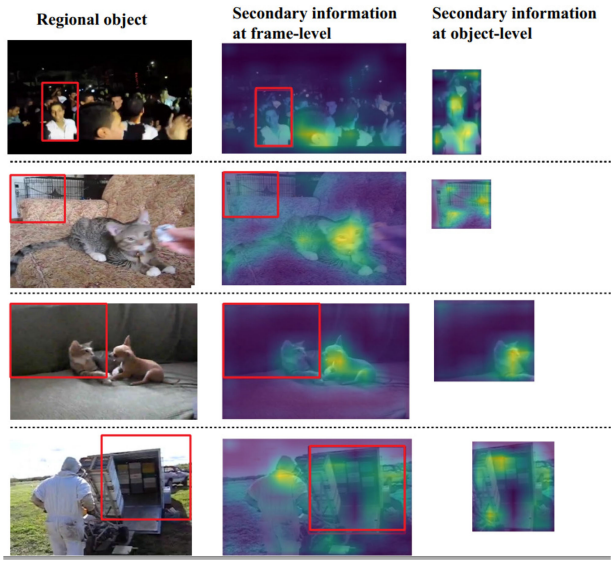


Fig. 4. Visualizing the neural activations of RoIs with heat map: (left) original frame with RoI; (middle) feature map of RoI; (right) feature map of scaled RoI. RoIs are surrounded by the red bounding box. Higher neural responses are shown in yellow color. This figure is best viewed in color.

IV. Object-VLAD

Object-VLAD encodes regional objects in a compact form for event classifier learning, while allowing flexible decoding of them into event evidences for recounting. Fig. 5 depicts the flowchart for generation of the Object-VLAD feature, which encodes the activations of DCNN for every frame and RoI extracted from a video. Frame-level activation refers to the feature corresponding to primary object or scene in a frame, while RoI-level activation refers to the object candidate regions sampled from video frames. These features are pooled temporally and spatially, and then encoded with VLAD into a compact video descriptor.

A. Primary Object or Scene

1) Visual Descriptors: Visual content can be characterized by neural activations. Similar to other approaches [14], two types of neural responses are respectively extracted from the feature map of the convolutional layer or the vector of the fully connected layer. We use pool5 to denote the last convolutional layer, and fc6 (fc7) as the first (second) fully connected layer. The fc feature undergone rectified linear unit (ReLU) is further named as fc_relu. Fig. 6 depicts the extraction of deep features. Pool5 (or feature map) is regarded as latent concept descriptor (LCD) [14] that spatially corresponds to the receptive fields of a video frame. In Object-VLAD, spatial pyramid pooling (SPP) is applied to LCD to generate LCD-SPP feature, which is empirically verified in [14], [55] as a more discriminative feature than LCD. As shown in Fig. 6, four different resolutions of spatial pooling is performed to generate a total of 50 feature vectors from the 7×7 feature map of VGG-19. On the other hand, fc6_relu feature is directly extracted from DCNN. Finally, a frame can either be depicted as a bag of LCD-SPPs (50 vectors) or a single fc vector. Thereby, by temporally collecting frame features, a video v can be represented by the two types of feature as following

$$v = \begin{cases} \{\boldsymbol{x}_i\}_{i=1,..I} & \text{fc} \\ \{\boldsymbol{x}_{i,j}\}_{i=1,..I;j=1,..J} & \text{LCD} - \text{SPP} \end{cases} \quad (1)$$

where $\boldsymbol{x} \in \mathbb{R}^D$ denotes feature in D dimension, i is index for frame and j is for LCD-SPP vector. The value of D depends on the type of DCNN. For example, D = 512 for LCD-SPP feature from VGG-19 and D = 2,048 for LCD-SPP feature from ResNet-50.

2) VLAD Encoding: Same as [14], LCD-SPP and fc vectors are respectively encoded by VLAD into two bags-of-features. For simplicity, we only describe the VLAD encoding for fc features. First, a fc vector $\boldsymbol{x}_i$ is transformed by PCA (principal components analysis) with whitening to a compact representation $\hat{\boldsymbol{x}}_i \in \mathbb{R}^d$ in $d < D$ dimensional space. A video descriptor v is then represented as a sequence of visual features $\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{x}}_i\}_{i=1}^N$. The sequence is encoded with VLAD dictionary, denoted as $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K\}$, where $\boldsymbol{\mu}_k \in \mathbb{R}^d$ is the k-th cluster centroid learned using K-means algorithm. VLAD associates each feature $\hat{\boldsymbol{x}}_i \in \hat{\boldsymbol{X}}$ of v to a centroid $\boldsymbol{\mu}_k$. The association can be based on hard or soft assignment, and the strength of association, denoted as $q_{i,k}$, is restricted by $q_{i,k} > 0$ and $\sum_{k=1}^K q_{i,k} = 1$. VLAD encodes the residuals between $\hat{\boldsymbol{x}}_i$ and $\boldsymbol{\mu}_k$ as

$$\boldsymbol{r}_k = \sum_{i=1}^{I} q_{i,k}(\hat{\boldsymbol{x}}_i - \boldsymbol{\mu}_k) \quad (2)$$

The VLAD vector is formed by stacking the residuals of centroids as following

$$\Phi(\hat{\boldsymbol{X}}) = [\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_k, \ldots, \boldsymbol{r}_K] \quad (3)$$

To this end, the video descriptor $\hat{\boldsymbol{X}}$ is encoded as a $d \cdot K$ dimensional histogram $\Phi(\hat{\boldsymbol{X}})$ for characterizing primary objects or scenes across the temporal axis.

B. Regional Objects

Despite being potential evidences of a multimedia event, regional objects (or RoIs) are presumably smaller in size, obscure in position, and likely to be overwhelmed by primary DCNN activation. Instead of extracting them as the "scaled feature map of RoIs", which will result in loss of information, Object-VLAD extracts and encodes "feature map of scaled RoI" as discussed in Section III.

1) Region Proposal: Objects can have varying sizes and appear at any locations of an image. Object localization can be conducted by windowing techniques [56], [57], which brute forcely input image patches scanned by multi-scale windows into DCNN. Although effective, these approaches are computationally expensive. More sophisticated techniques such as Fast-RCNN [51], which requires supervised learning, can propose and classify candidate regions. However, these models are generally limited by the number of object categories available for training. For example, Fast-RCNN can only scale up to 200 pre-trained categories. Unsupervised algorithms, which proposes objects based on local visual properties, are more appropriate. As video frames suffer from motion blur, we employ selective

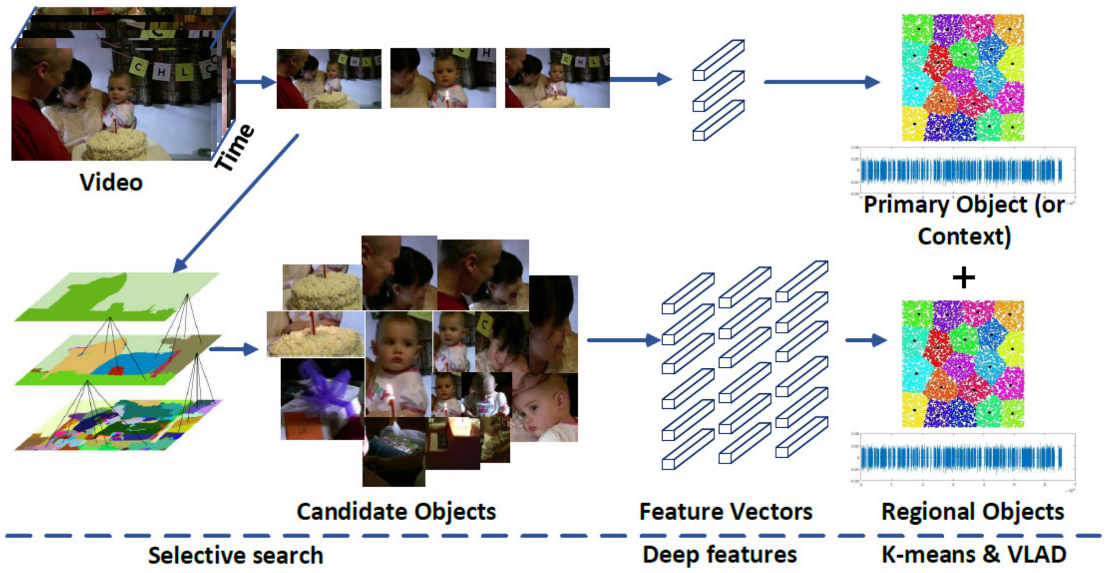


Fig. 5. Overview of Object-VLAD. The inputs to DCNN are frames and object candidates proposed by selective search. The extracted deep features represent the responses from primary object/scene and regional objects respectively, which are encoded with VLAD learnt via K-means. The figure is best viewed in color.

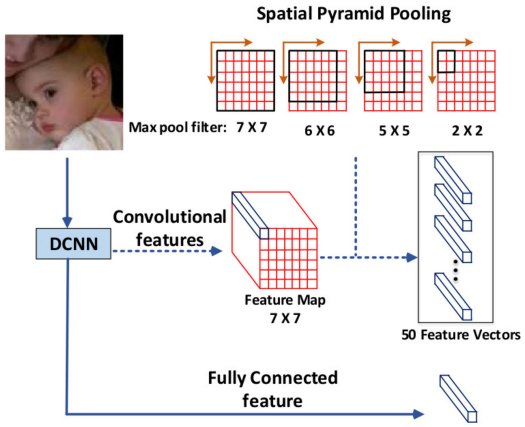


Fig. 6. Deep feature repesentation type. 1) a feature vector from fully connected layer (solid line), 2) 50 feature vectors from spatial pyramid pooling of feature maps (dashed line).

search [58] that exploits color and brightness than the edge and saliency-based methods such as BING [59], objectness [60] and edge box [61].

2) Visual Descriptors & VLAD Encoding: Each extracted proposal undergoes the same procedure as equations (1)–(3) for VLAD encoding. Denote the resulting video descriptor as $\hat{\boldsymbol{Y}} = \{\hat{\boldsymbol{y}}_{i,n}\}$, where $\hat{\boldsymbol{y}}_{i,n} \in \mathbb{R}^d$ is the d-dimensional visual feature of n proposal at i-th video frame. Further denote I and N as the number of frames and proposals in a video respectively, the VLAD encoded vector is in the form of

$$\boldsymbol{l}_k = \sum_{i=1}^{I}\sum_{n=1}^{N} q_{i,n,k}(\hat{\boldsymbol{y}}_{i,n} - \boldsymbol{\rho}_k) \tag{4}$$

$$\Phi(\hat{\boldsymbol{Y}}) = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_k, \ldots, \boldsymbol{l}_K] \tag{5}$$

where $\boldsymbol{\rho}_k$ represents the k-th cluster centroid of VLAD dictionary and $q_{i,n,k}$ quantifies the association between $\hat{\boldsymbol{y}}_{i,n}$ and $\boldsymbol{\rho}_k$.

Finally, Object-VLAD is formed by concatenating the VLAD vectors of primary and regional objects as following

$$\boldsymbol{H} = [\Phi(\hat{\boldsymbol{X}}), \Phi(\hat{\boldsymbol{Y}})] \tag{6}$$

V. Event Detection and Recounting

Using Object-VLAD as the features of training examples, multimedia event detection (MED) can be straightforwardly performed by classifier learning. We employ linear SVM for its efficiency in classification. Denote $\boldsymbol{W}$ as the hyperplane learned by SVM, the decision function of an input video feature $\boldsymbol{H}$ is

$$f(\boldsymbol{H}) = \boldsymbol{W}^T \boldsymbol{H} \tag{7}$$

In equation (7), while $\boldsymbol{H}$ captures the distributions of primary and regional objects in a video, the hyperplane anchors the significances of these distributions. The main idea of multimedia event recounting (MER) is by "unrolling" $\boldsymbol{H}$ and capitalizing on $\boldsymbol{W}$ to rank the significances of RoIs. Ideally, the top-rank RoIs serve as evidences to justify the presence of event. The unrolling of $\boldsymbol{H}$ is achieved by first substituting equation (6) into equation (7), resulting in

$$f(\boldsymbol{H}) = \boldsymbol{W_x}^T \Phi(\hat{\boldsymbol{X}}) + \boldsymbol{W_y}^T \Phi(\hat{\boldsymbol{Y}}) \tag{8}$$

where $\boldsymbol{W_x}$ and $\boldsymbol{W_y}$ are sub-vectors of $\boldsymbol{W}$ corresponding to $\Phi(\hat{\boldsymbol{X}})$ and $\Phi(\hat{\boldsymbol{Y}})$ respectively. By combining equations (2)–(5) into equation (8), the decision function is rewritten as

$$f(\boldsymbol{H}) = \sum_{k=1}^{K}\sum_{i=1}^{I} q_{i,k} \boldsymbol{W_{x,k}}^T (\hat{\boldsymbol{x}}_i - \boldsymbol{\mu}_k) + \sum_{k=1}^{K}\sum_{i=1}^{I}\sum_{n=1}^{N} q_{i,n,k} \boldsymbol{W_{y,k}}^T (\hat{\boldsymbol{y}}_{i,n} - \boldsymbol{\rho}_k) \tag{9}$$

$$= \sum_{i=1}^{I} C(\hat{\boldsymbol{x}}_i) + \sum_{i=1}^{I}\sum_{n=1}^{N} C(\hat{\boldsymbol{y}}_{i,n}) \tag{10}$$

TABLE I
THE 30 EVENTS DEFINED IN TRECVID MED 14 AND MED 13 DATASETS

MED13		MED13 & MED14		MED14	
ID	Event Name	ID	Event Name	ID	Event Name
E006	Birthday party	E021	Attempting a bike trick	E031	Beekeeping
E007	Changing a vehicle tire	E022	Cleaning an appliance	E032	Wedding shower
E008	Flash mob gathering	E023	Dog show	E033	Non-motorized vehicle repair
E009	Getting a vehicle unstuck	E024	Giving directions to a location	E034	Fixing musical instrument
E010	Grooming animal	E025	Marriage proposal	E035	Horse riding competition
E011	Making a sandwich	E026	Renovating a home	E036	Felling a tree
E012	Parade	E027	Rock climbing	E037	Parking a vehicle
E013	Parkour	E028	Town hall meeting	E038	Playing fetch
E014	Repairing an appliance	E029	Winning a race without a vehicle	E039	Tailgating
E015	Working on a sewing project	E030	Working on a metal crafts project	E040	Tuning a musical instrument

where I and N are the numbers of frames and object proposals respectively. The function $C(\cdot)$ quantifies object significance as following

$$C(\hat{\boldsymbol{x}}_i) = \sum_{k=1}^{K} q_{i,k} \boldsymbol{W_{x,k}}^T (\hat{\boldsymbol{x}}_i - \boldsymbol{\mu}_k) \tag{11}$$

$$C(\hat{\boldsymbol{y}}_{i,n}) = \sum_{k=1}^{K} q_{i,n,k} \boldsymbol{W_{y,k}}^T (\hat{\boldsymbol{y}}_{i,n} - \boldsymbol{\rho}_k) \tag{12}$$

where K corresponds to the size of VLAD dictionary, $\boldsymbol{W_{x,k}}$ is the sub-vector of $\boldsymbol{W_x}$ referring to k-th cluster centroid, and similarly for $\boldsymbol{W_{y,k}}$. By further combining equations (11) and (12), the contribution of a frame is measured as

$$C(f_i) = C(\hat{\boldsymbol{x}}_i) + \sum_{n=1}^{N} C(\hat{\boldsymbol{y}}_{i,n}) \tag{13}$$

In short, equation (10) decomposes $\boldsymbol{H}$ into primary and regional objects, each weighted by either equation (11) or (12). In this way, MER can prompt different level of evidence, frames as thumbnails and RoIs as object evidences, by sorting their respective contributions to the presence of an event.

VI. EXPERIMENTAL SETTINGS

A. Datasets

Experiments are conducted on TRECVID MED14 [10] and MED13 datasets [62]. Table I shows the 20 multimedia events on each dataset, with 10 overlapping events between them. There are three subsets for each dataset: training, background and testing sets, where the former two sets are for model learning and validation. The training set contains 3,000 videos, including 100 positives examples per event. Background set has 5,000 videos which may or may not be relevant to the twenty defined events. The test set contains 30,000 videos for evaluation purpose. Throughout the subsets, keyframes are uniformly sampled at the rate of 1 frame per two seconds. The average length of a video is 2.4 minutes and with 80 keyframes.

B. Settings

1) Regions: Selective search [58] is employed to propose object regions. "Single strategy" is used to sample around 200 candidates from a keyframe. To avoid costly processing of all the proposals, a number of criteria is defined for rapid filtering of false positives. The criteria include pruning regions of tiny size or elongated shape, and minimize the total overlapping area among the selected proposals. Specifically, tiny regions with areas smaller than 10% of the original frame are removed. The allowable ratio between two regions (b_1, b_2) is kept to be $overlap = \frac{area(b_1 \cap b_2)}{area(b_1 \cup b_2)} < 0.5$. With these criteria, the average number of proposals for each keyframe is 20. These regions are subsequently resized into a fixed resolution of 227×227 before feeding into VGG-19 by Caffe [63] for feature extraction.

2) VLAD: The encoding of video features follows the optimal setting suggested by [14]. The deep feature, either extracted from fc6_relu or LCD-SPP, is first reduced to a 256-dimensional vector by PCA with whitening. The VLAD dictionary is learned on a dataset with 520,000 features randomly sampled from training and background sets. K-means is employed to cluster the features into 256 clusters. The implementation of VLAD encoding is based on VLFeat toolbox [64].

3) Evaluation: Two scenarios, learning with abundant (100Ex) and very few (10Ex) training examples, are considered. In 100Ex setting, a total of 100 positive examples per event is provided for classifier learning versus 10 positive examples in 10Ex.

4) Event Classifier Learning: Linear SVM, provided by LIBSVM toolbox [65], is employed for classifier training. The penalty parameter C in SVM is determined via 5-fold cross-validation. The same value of C is used in both 100Ex and 10Ex tasks.

VII. RESULTS AND DISCUSSION

A. Event Detection

All the DCNN models, such as AlexNet, VGG-19, ResNet-50, are pre-trained on ImageNet 1000 dataset [66]. The performance evaluation is based on mean average precision (mAP). Fig. 7 contrasts the per event mAP of Object-VLAD with CNN-VLAD baseline, which encodes only frame-level features, on the MED14 dataset. When using fc6_relu as the feature, Object-VLAD outperforms baseline for 19 out of 20 events in both 100Ex and 10Ex tasks. The relative improvements are 14.5% and 27.6% respectively. A similar level of improvement is also noted when using LCD-SPP as the features.

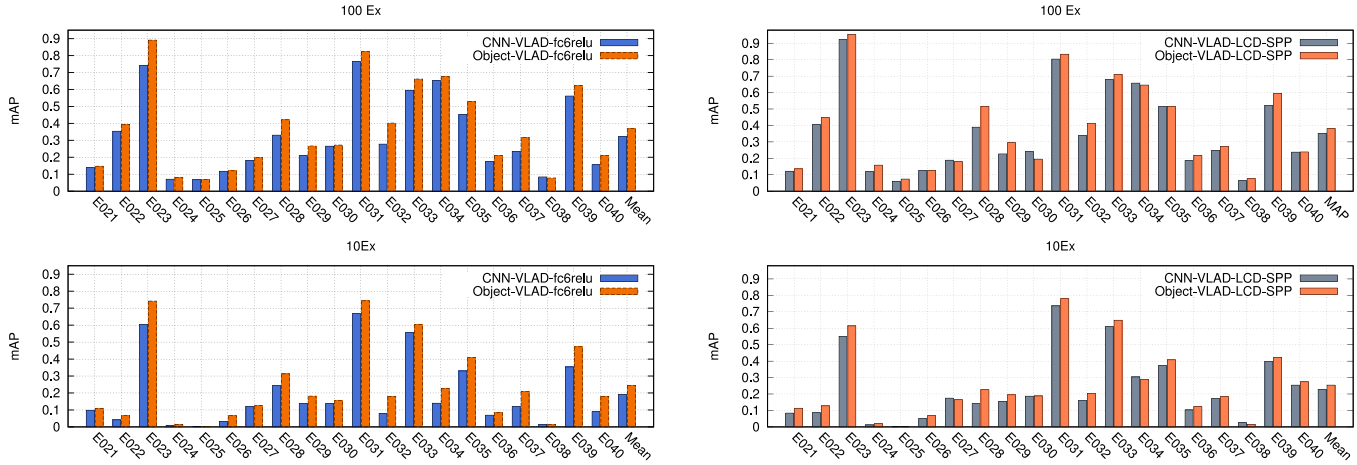
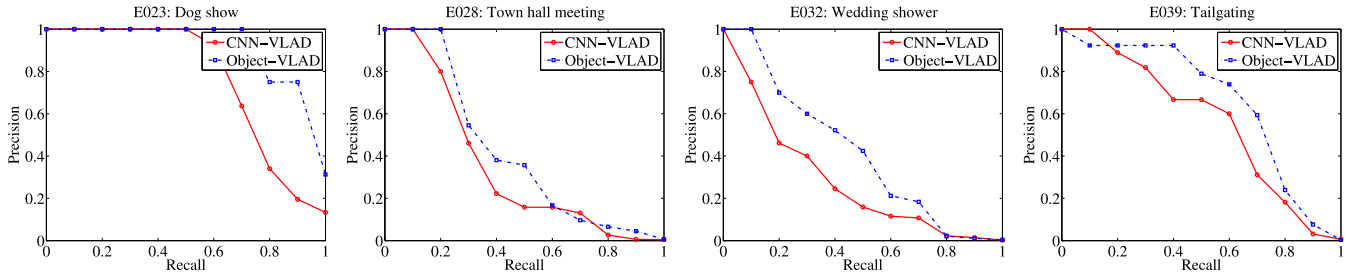
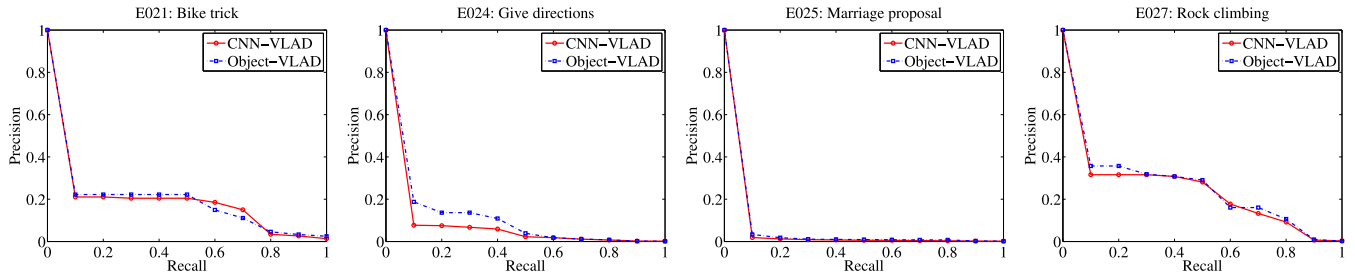


Fig. 7. MED14 100Ex/10Ex per event performance comparison with fc6_relu feature (**left**) and LCD-SPP feature (**right**).



(a) Four events with relatively ***large*** mAP improvement.



(b) Four events with ***insignificant*** mAP improvement.

Fig. 8. Interpolated precision-recall curves for 100Ex task on MED14 dataset using fc6_relu features.

To further provide empirical insights, Fig. 8 shows the 11-point interpolated precision-recall curves for events, where Object-VLAD attain relative **large** (a) and **minor** (b) improvement. For events such as "dog show" (E023), the scene is often cluttered with objects such as "dog", "fence" and "carpet". While CNN-VLAD tends to overlook scene content for these events, Object-VLAD manages to capture and encode some of these objects as features and thus exhibits a better performance. In contrast, for events involving mainly single object such as "bicycle" in "bike trick" (E021) or object in small size such as "engagement ring" in "marriage proposal" (E025), there is no obvious difference between Object-VLAD and CNN-VLAD. As observed, Object-VLAD is capable of boosting the ranking of positive videos when relevant objects are sampled for VLAD encoding (Fig. 8(a)), while not adversely impact performance even if irrelevant regions are included (Fig. 8(b)).

To verify that the improvement introduced by Object-VLAD is not by chance, we employ partial randomization test to compare per event query performance obtained by Object-VLAD and the baseline. Detail settings include 20,000 random permutations, with a **null** hypothesis that the improvement is brought by chance. Randomization test shows that Object-VLAD is significantly different from CNN-VLAD at the level of $p = 0.05$, where the $p-$value is close to 0 indicates the rejection of null hypothesis. The tool[1] of randomization test is provided by TRECVID organizer.

B. Impacts of Visual Descriptors

Next, we study the robustness of Object-VLAD when fc7_relu and LCD-SPP features are in use. The features are extracted from VGG-19 and ResNet-50 architectures. Table II lists

[1]http://www.nlpir.nist.gov/projects/t01v/trecvid.tools/randomization.testing/

TABLE II
PERFORMANCE COMPARISON (MAP) USING DIFFERENT VISUAL DESCRIPTORS

(a) MED14-100Ex

	fc7_relu (VGG-19)	fc6_relu (VGG-19)	LCD-SPP (VGG-19)	LCD-SPP (ResNet-50)
CNN-VLAD [14]	0.315	0.326	0.357	-
Our implementation	0.311	0.322	0.353	0.386
Object-VLAD	**0.364**	**0.369**	**0.380**	**0.40**

(b) MED14-10Ex

	fc7_relu (VGG-19)	fc6_relu (VGG-19)	LCD-SPP (VGG-19)	LCD-SPP (ResNet-50)
CNN-VLAD [14]	-	-	0.232	-
Our implementation	0.175	0.192	0.229	0.253
Object-VLAD	**0.229**	**0.245**	**0.254**	**0.272**

TABLE III
MED14-100EX: PERFORMANCES OF ENCODING FRAMES, REGIONS AND BOTH

	CNN-VLAD	Region-VLAD	Object-VLAD
MED14-100Ex	0.353	0.369	**0.380**
MED14-10Ex	0.229	0.253	**0.254**

TABLE IV
COMPARISON OF MULTI-FEATURES FUSION ON MED14 DATASET

	MED14-100Ex	MED14-10Ex
CNN-VLAD (fc6, fc7, LCD-SPP) [14]	0.368	0.245
Our implementation (fc6_relu, LCD-SPP)	0.354	0.235
Object-VLAD (fc6_relu, LCD-SPP)	**0.387**	**0.268**

the performance of Object-VLAD in comparison to two implementations of CNN-VLAD. Basically, the performance keeps improving with better features (LCD-SPP) and more powerful architectures (ResNet-50). Object-VLAD consistently outperforms baselines across features and architectures in both 100Ex and 10Ex tasks. For VGG-19, a larger degree of improvement is noted for Object-VLAD when using fc6_relu and fc7_relu than LCD-SPP.

C. Impacts of Regional Objects

We further assess the contribution of Object-VLAD in encoding primary scene (CNN-VLAD) and regional objects (Region-VLAD). Based on LCD-SPP features, Table III shows the contributions of each component in Object-VLAD. For the 100Ex and 10Ex tasks, Region-VLAD exhibits higher mAP than CNN-VLAD. As both features are complementary to each other, overall higher mAP is attained for 100Ex and 10Ex when both components are fused.

D. Results for Multiple Features Fusion

Fusing multiple features are likely to introduce further improvement and Table IV lists the performance. The CNN-VLAD implementation in [14] reports improvement with the late fusion of three different features. The improvement, nevertheless, is not higher than Object-VLAD even when using only fc6_relu or LCD-SPP. For Object-VLAD, further but less significant improvement in mAP is noticed when fc6_relu and LCD-SPP are used. In short, encoding of regional objects appears more critical in boosting event detection performance than multi-feature fusion.

TABLE V
COMPARISON OF OBJECT-VLAD WITH THE OTHER VIDEO REPRESENTATIONS

	MED14 100Ex	MED14 10Ex	MED13 100Ex	MED13 10Ex
IDT-FV [6]	0.255	0.122	0.326	-
VideoStory [27]	-	-	-	0.196
VideoStory [24]	0.259	-	-	-
BoF [24]	0.276	-	-	-
Object Pooling [29]	0.312	0.193	-	-
DevNet (SVM) [46]	0.329	-	-	-
DevNet (KR) [46]	0.333	-	-	-
CNN-VLAD [14]	0.357	0.232	0.403	0.256
Our Implementation (LCD-SPP)	0.353	0.229	0.425	0.272
Object-VLAD (LCD-SPP)	**0.380**	**0.254**	**0.45**	**0.292**

E. Analysis of Speed Efficiency

Unlike CNN-VLAD that operates directly on frame-level features, Object-VLAD quantizes fine-grained regional-level features. As the Object-VLAD trades computational cost for detection performance, speed efficiency is undoubtedly degraded. The longer processing speed is caused by region proposal using selective search and repeated extraction of regional feature for each proposal. On a Tesla K40 GPU, Object-VLAD is about 24 times slower than CNN-VLAD. For a video of 80 sampled frames, Object-VLAD takes 513 seconds versus CNN-VLAD that requires 21 seconds for feature extraction.

To investigate the trade-off between speed and retrieval accuracy, we implement a variant of Object-VLAD based on "scaled feature map of RoI" using R-FCN [67]. As illustrated in Fig. 2, the neural activations corresponding to RoIs are extracted from feature map for VLAD encoding. The R-FCN is trained with ResNet-50 backbone using 818 object categories. With this implementation, Object-VLAD is sped up by 18 times at the expense of retrieval accuracy. Using LCD-SPP as the feature, the mAP drops from 0.40 to 0.392 for MED14-100Ex and from 0.272 to 0.270 for MED14-10Ex.

F. Comparison to the Other Methods

Next, we compare Object-VLAD with several state-of-the-art results reported on MED14 and MED13 datasets. As listed in Table V, except IDT-FV [6] and VideoStory [27] which are based on hand-crafted motion features, the remaining approaches are based on deep features extracted from DCNN. These approaches cover a wide variety of strategies in encoding visual features for boosting detection performance. VideoStory utilizes embedding features learnt jointly by motion features and textual video descriptions. BoF [24] smartly selects discriminative fragments from training videos to learn the bag-of-fragments dictionary for feature encoding. Object Pooling [29] represents object regions as histograms of concept responses and max pools the histograms as video descriptor. DevNet [46], on the other hand, explores the correlation among 20 events defined in MED14 dataset for learning better frame-level features that ideally also

encode the objects salient to events. Using the features, two separate classifiers, SVM and KR (kernel ridge regression) are then trained for event detection. Different from these approaches, Object-VLAD treats each event independently, directly exploits regional cues and encodes features with VLAD. For fair comparison, both Object-VLAD and CNN-VLAD use LCD-SPP as visual descriptor.

As shown in Table V, Object-VLAD achieves the overall best performance for both 100Ex and 10Ex tasks on MED datasets. The advantage of VLAD is evidenced by a large improvement when compared to Object Pooling [29] which encodes regional features with max pooling. More importantly, the result empirically confirms that the merit of encoding object information is more than the strategies that exploit multi-modal features (VideoStory) [27] and temporally significant fragments (BoF) [24]. Even compared to DevNet [46] which strives to learn features relevant to object evidences by exploring inter-event correlation, Object-VLAD still exhibits much better performance. Furthermore, BoF and DevNet, different from Object-VLAD, are not applicable to the 10Ex task requiring sufficient number of training samples to learn discriminative features.

TABLE VI
COMPARISON OF EVENT RECOUNTING RESULTS WITH DIFFERENT VIDEO REPRESENTATIONS (NAP/AIR)

	Non-evidenced frames (%)	ISOMER(%)	CNN-VLAD (%)	ISOMER-Object(%)	Object-VLAD(%)
E021	43.8	60.0 / 31.7	61.5 / 31.2	**62.0** / **31.1**	58.6 / 31.3
E022	43.1	59.5 / 33.8	58.7 / 33.2	59.6 / 33.6	**66.5** / **32.1**
E023	23.3	49.9 / 14.3	53.9 / 14.4	54.9 / **14.0**	**58.9** / **14.0**
E024	79.0	79.1 / 79.2	81.5 / 78.3	79.5 / 76.9	**83.9** / **76.3**
E025	69.8	65.9 / 70.7	**72.6** / 67.2	68.9 / 69.2	**72.3** / **67.0**
E026	13.2	40.4 / -	44.8 / -	42.9 / -	**55.7** / -
E027	7.9	22.0 / -	23.7 / -	20.2 / -	**24.5** / -
E028	9.7	35.0 / -	68.0 / -	42.2 / -	**80.9** / -
E029	31.0	54.1 / 21.8	54.2 / 21.0	57.8 / 21.2	**58.4** **20.9**
E030	6.0	10.1 / -	15.2 / -	11.1 / -	**28.1** / -
E031	20.5	51.7 / 12.4	60.1 / 12.1	55.0 / 12.2	**67.6** / **11.7**
E032	31.5	38.4 / 25.9	38.4 / 25.0	40.7 / 25.9	**42.8** / **23.9**
E033	9.5	26.0 / -	44.1 / -	24.5 / -	50.9 / -
E034	10.8	64.9 / -	50.4 / -	68.6 / -	**72.9** / -
E035	14.7	38.3 / -	36.2 / -	46.0 / -	**54.1** / -
E036	23.4	**43.3** / 16.3	36.6 / 16.5	42.5 / 16.3	38.9 / **16.0**
E037	12.0	20.1 / -	25.5 / -	19.6 / -	**31.0** / -
E038	32.8	40.2 / 25.0	66.8 / 24.7	49/.0 / 23.5	**70.6** / **23.3**
E039	6.6	30.0 / -	29.9 / -	30.6 / -	**38.0** / -
E040	12.1	38.5 / -	38.7 / -	35.2 / -	**49.0** / -
Mean	25.0	43.4 / 33.1	48.0 / 32.3	45.5 / 32.4	**55.2** / **31.6**

G. Event Recounting

Multimedia event recounting (MER) summarizes a retrieved video by presenting visual snippets composed of frames or clips as evidences. In TRECVID benchmarking activity [68], the evaluation is conducted by recruiting evaluators to score the quality of visual snippets. In this experiment, our purpose is to assess the relevancy of snippets, rather than the quality in numerical score, in explaining the presence of events. As the relevance can be objectively judged with binary decision (yes or no), we manually label the keyframes of each positive videos as evidential or non-evidential frames. With this, we propose two new measures named negative average precision (NAP) and average irrelevant rate (AIR) for evaluation. NAP assesses the average precision in which non-evidential frames are ranked, while AIR measures the error rate in suggesting non-evidence frames. The input to these measures is a rank list of video frames sorted in descending order based on their contributions to the presence of an event. Denote tn as the tn-th non-evidential frame and $rank(tn)$ as the position of tn in the list of length l. Then NAP and AIR are defined as below to assess the quality of the sorted list:

$$NAP = \frac{1}{|M|}\sum_{tn=1}^{M}\frac{M - tn}{l - rank(tn)} \tag{14}$$

$$AIR = \frac{1}{|M|}\sum_{tn=1}^{M}\frac{tn}{rank(tn)} \tag{15}$$

where M denotes the number of non-evidential frames in a positive video.

Note that higher value of NAP and lower value of AIR indicate better recounting capability. Basically, NAP imposes punishment when evidential frames are ranked at the bottom of the list, and AIR penalizes the list where non-evidential frames are ranked high. To construct ground-truth for evaluation, we label the keyframes of a positive video as either evidential or non-evidential. The former refers to the case that a keyframe contains objects that could evidence the presence of an event, and vice versa.

We compare the performance of Object-VLAD with CNN-VLAD and ISOMER [44]. Object-VLAD sorts evidential frames based on Equation (13), and similarly for CNN-VLAD based on Equation (11). Both approaches use fc6_relu features extracted from VGG-19. For ISOMER, the video descriptor is generated based on average pooling of frame-level features, which are histograms of concept response. For fair comparison, the features for ISOMER are also extracted from VGG-19 and the event classifiers are trained based on linear SVM. To demonstrate the power of encoding regional objects, we also implement a variant named ISOMER-Object, which similar to Object-VLAD employs selective search for region proposals and averagely pools the fc6_relu feature of proposals as video descriptor. The implementations are detailed below.

- ISOMER[44]: The video descriptor is averagely pooled from frame-level features $\hat{\boldsymbol{x}}_i$ as:

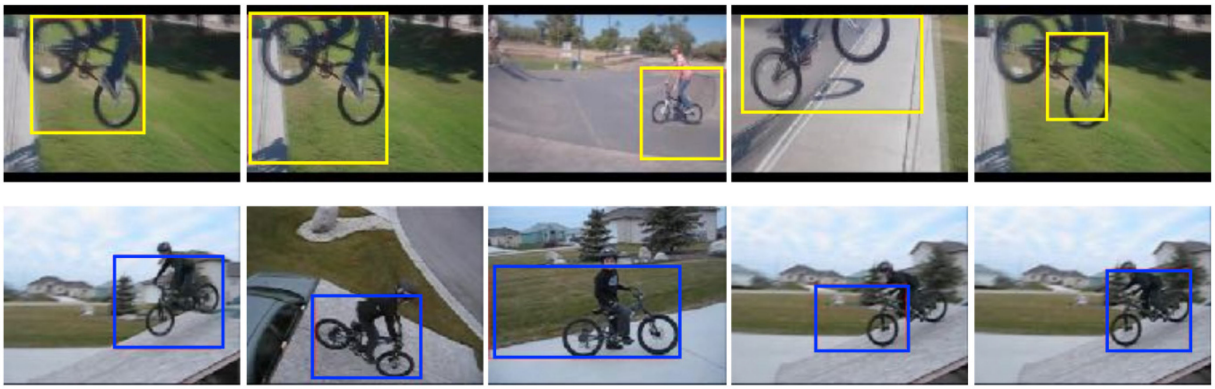
$$\hat{\boldsymbol{X}}_{AVE} = \frac{1}{I}\sum_{i=1}^{I}\hat{\boldsymbol{x}}_i \tag{16}$$

where I is the number of frames. Given a video descriptor, the decision function of a linear SVM classifier is

$$f(\hat{\boldsymbol{X}}_{AVE}) = W^T\hat{\boldsymbol{X}}_{AVE} = \sum_{i=1}^{I}\frac{1}{I}W^T\hat{\boldsymbol{x}}_i \tag{17}$$

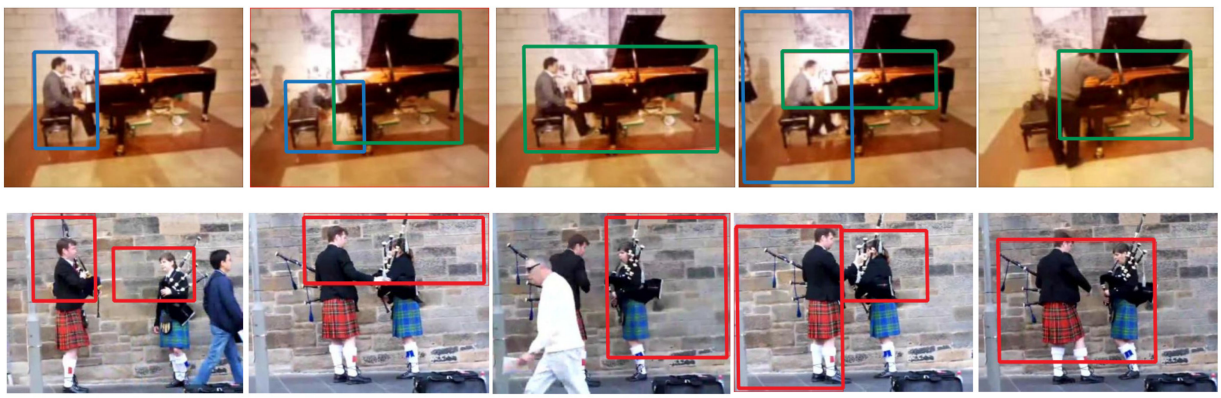
and the contribution score $C(f_i)$ of a frame f_i is calculated as:

$$C(f_i) = \frac{1}{I}W^T\hat{\boldsymbol{x}}_i \tag{18}$$



(a) Bike Trick (E021): the suggested evidences are the "bike".

(b) Dog Show (E023): a number of object evidences such as “dog" and “fence" are suggested.



(c) Tuning a musical instrument (E040): a number of object evidences such as “person" and “musical instrument" are suggested.

Fig. 9. Examples of top-5 evidences for three different events. Each row shows a video and the evidences are sorted from left to right. The figure is best viewed in color.

With sigmoid function, $C(f_i)$ is further transformed into a confidence score $P(f_i)$:

$$P(f_i) = \frac{1}{1 + e^{-I \cdot C(f_i)}} \propto C(f_i) \tag{19}$$

- ISOMER-Object: The video descriptor $\boldsymbol{H}$ is composed of two parts with average pooling of frame-level and region-level features respectively, as following:

$$\boldsymbol{H} = [\hat{\boldsymbol{X}}_{AVE}, \hat{\boldsymbol{Y}}_{AVE}] \tag{20}$$

$$= \left[\sum_{i=1}^{I} \frac{1}{I} \hat{\boldsymbol{x}}_i, \sum_{i=1}^{I} \sum_{n=1}^{N} \frac{1}{NI} \hat{\boldsymbol{y}}_{i,n} \right] \tag{21}$$

where N is the number of regions. With linear SVM, the decision function is:

$$f(\boldsymbol{H}) = \boldsymbol{W_x}^T \sum_{i=1}^{I} \frac{1}{I} \hat{\boldsymbol{x}}_i + \boldsymbol{W_y} \sum_{i=1}^{I} \sum_{n=1}^{N} \frac{1}{NI} \hat{\boldsymbol{y}}_{i,n} \tag{22}$$

$$= \sum_{i=1}^{I} \left(\boldsymbol{W_x}^T \frac{1}{I} \hat{\boldsymbol{x}}_i + \boldsymbol{W_y} \sum_{n=1}^{N} \frac{1}{NI} \hat{\boldsymbol{y}}_{i,n} \right) \tag{23}$$

and the contribution score of a frame is calculated as:

$$C(f_i) = \boldsymbol{W_x}^T \frac{1}{I} \hat{\boldsymbol{x}}_i + \boldsymbol{W_y} \sum_{n=1}^{N} \frac{1}{NI} \hat{\boldsymbol{y}}_{i,n} \tag{24}$$

Same as (19), $C(f_i)$ is transformed into a confidence score $P(f_i)$.

The experiment is conducted on MED14 dataset for 100Ex task and the result is listed in Table VI for NAP and AIR. For AIR, we exclude the events whose percentage of non-evidential frames is less than 15%. For these events, majority of frames are evidential and there is no significant difference among these approaches. Basically, for all the approaches, the performance is inversely proportional to the percentage of non-evidential frames. For NAP, we consider all the events since the measure assesses the ability of ranking evidential frames as high as possible. Different from AIR, the performance generally decreases as the percentage of non-evidential frames reduces.

In both NAP and AIR, as evidenced by both Object-VLAD and ISOMER-Object, object-level encoding leads to consistent improvement over frame-level encoding. In addition, VLAD encoding also exhibits better performance than average pooling. Overall Object-VLAD attains the best performance among the four approaches. The degree of improvement in terms of NAP is noticeably larger than AIR, which basically hints the better capability of Object-VLAD in ranking evidential frames. While Table VI assess the frame-level evidence, Fig. 9 lists some examples of object evidences recounted by Object-VLAD. For some events such as "bike trick" (E021), there is only one type of prominent object evidence. Reversely, for events such as "dog show" (E023) where the key object "dog" is in small size, more object evidences will be suggested. Additionally, for complex events like "tuning a musical instrument" (E40), objects including performers and instruments are proposed as evidences. In this event, frame-level interaction between person and instrument is frequently observed, and equation (12) assigns high weights to both objects for recounting.

To verify that the performance improvement is not by chance, we conduct significance tests using the same settings as Section VII-A for both NAP and AIR. First, the performance of Object-VLAD is verified to be significantly different from CNN-VLAD at $p = 0.05$, showing the contribution of object-level representation. Second, VLAD decoding also exhibits significantly better performance at $p = 0.05$, when comparing Object-VLAD to ISOMER-Object and CNN-VLAD to ISOMER.

VIII. Conclusion

We have presented Object-VLAD in overcoming the drawback of DCNN features. We show the effectiveness of keeping secondary object information, by incorporating object proposal, DCNN mid-level feature and VLAD encoding into a pipeline for MED and MER. Consistent improvement of Object-VLAD over different branches of approaches empirically verifies our claim that recovering the "loss-in-details" in DCNN is essential for multimedia event description. Currently, Object-VLAD does not consider mining of event-discriminative fragments and objects during VLAD encoding, which has been verified in [24] as critical in excluding irrelevant clips from event classifier learning. Future direction includes incorporation of visual attention model such as in [69] into feature encoding. In addition, scarcity in positive training examples for multimedia event detection is always a challenge. Active learning strategies such as [70], [71] are promising direction to enhance the robustness of detection.

References

[1] R. Zhao and W. I. Grosky, "Narrowing the semantic gap-improved text-based web document retrieval using visual features," *IEEE Trans. Multimedia*, vol. 4, no. 2, pp. 189–200, Jun. 2002.

[2] A. Hauptmann, R. Yan, W.-H. Lin, M. Christel, and H. Wactlar, "Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 958–966, Aug. 2007.

[3] H. Ma, J. Zhu, M. R.-T. Lyu, and I. King, "Bridging the semantic gap between image contents and tags," *IEEE Trans. Multimedia*, vol. 12, no. 5, pp. 462–473, Aug. 2010.

[4] A. Joly, O. Buisson, and C. Frelicot, "Content-based copy retrieval using distortion-based probabilistic similarity search," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 293–306, Feb. 2007.

[5] F. Wang, Z. Sun, Y.-G. Jiang, and C.-W. Ngo, "Video event detection using motion relativity and feature selection," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1303–1315, Aug. 2014.

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

[7] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[8] A. Karpathy *et al.*, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1725–1732.

[9] K. Soomro, A. Zamir, and M. Shah, "Ucf101: A dataset of 101 human action classes from videos in the wild," Center for Research in Computer Vision, Univ. Central Florida, Orlando, FL, USA, Tech. Rep. CRCV-TR-12-01, 2012.

[10] P. Over *et al.*, "TRECVID 2014–An overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proc. TREC Video Retrieval Eval.*, 2014, p. 52.

[11] X. Han, B. Singh, V. Morariu, and L. S. Davis, "VRFP: On-the-fly video retrieval using web images and fast fisher vector products," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1583–1595, Jul. 2017.

[12] P. Mettes, D. C. Koelma, and C. G. Snoek, "The imagenet shuffle: Reorganized pre-training for video event detection," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2016, pp. 175–182.

[13] M. Jain, J. C. van Gemert, and C. G. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 46–55.

[14] 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.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *CoRR*, vol. abs/1409.1556, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[18] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3304–3311.

[19] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 2, pp. 2169–2178.

[20] 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, no. 2, pp. 73–101, 2013.

[21] 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.

[22] M. Mazloom, A. Habibian, D. Liu, C. G. 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.

[23] K.-T. Lai, X. Y. Felix, M.-S. Chen, and S.-F. Chang, "Video event detection by inferring temporal instance labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2251–2258.

[24] P. Mettes, J. C. van Gemert, S. Cappallo, T. Mensink, and C. G. Snoek, "Bag-of-fragments: Selecting and encoding video fragments for event detection and recounting," in *Proc. 5th ACM Int. Conf. Multimedia Retrieval*, 2015, pp. 427–434.

[25] X. Chang, Y. Yang, E. P. Xing, and Y.-L. Yu, "Complex event detection using semantic saliency and nearly-isotonic SVM," in *Proc. 32th Int. Conf. Mach. Learn.*, 2015, pp. 1348–1357.

[26] X. Zhang *et al.*, "Enhancing video event recognition using automatically constructed semantic-visual knowledge base," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1562–1575, Sep. 2015.

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

[28] I.-H. Jhuo and D. Lee, "Video event detection via multi-modality deep learning," in *Proc. 22nd Int. Conf. IEEE Pattern Recognit.*, 2014, pp. 666–671.

[29] H. Zhang and C.-W. Ngo, "Object pooling for multimedia event detection and evidence localization," *ITE Trans. Media Technol. Appl.*, vol. 4, pp. 218–226, 2016.

[30] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, "Exploiting image-trained CNN architectures for unconstrained video classification," in *Proc. 26th Brit. Mach. Vis. Conf.*, 2015, pp. 60.1–60.13.

[31] 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.

[32] C. Ngo *et al.*, "VIREO-TNO@ TRECVID 2014: Multimedia event detection and recounting (MED and MER)," in *Proc. TREC Video Retrieval Eval.*, 2014.

[33] Y.-J. Lu, H. Zhang, M. de Boer, and C.-W. Ngo, "Event detection with zero example: Select the right and suppress the wrong concepts," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2016, pp. 127–134.

[34] 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, pp. 297–304.

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

[36] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5297–5307.

[37] Z. Zhao *et al.*, "BUPT-MCPRL at TRECVID 2016," in *Proc. TREC Video Retrieval Eval.*, 2016.

[38] J. Hou, X. Wu, F. Yu, and Y. Jia, "Multimedia event detection via deep spatial-temporal neural networks," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2016, pp. 1–6.

[39] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei, "You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 923–932.

[40] Z. Ma, X. Chang, Y. Yang, N. Sebe, and A. Hauptmann, "The many shades of negativity," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1558–1568, Jul. 2017.

[41] Y.-G. Jiang, Q. Dai, T. Mei, Y. Rui, and S.-F. Chang, "Super fast event recognition in internet videos," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1174–1186, Aug. 2015.

[42] C.-C. Tan and C.-W. Ngo, "On the use of commonsense ontology for multimedia event recounting," *Int. J. Multimedia Inf. Retrieval*, vol. 5, no. 2, pp. 73–88, 2016.

[43] D. Ding *et al.*, "Beyond audio and video retrieval: Towards multimedia summarization," in *Proc. 2nd ACM Int. Conf. Multimedia Retrieval*, 2012, pp. 2–9.

[44] C. Sun *et al.*, "Isomer: Informative segment observations for multimedia event recounting," in *Proc. Int. Conf. Multimedia Retrieval*, 2014, pp. 241–248.

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

[46] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann, "DEVNET: A deep event network for multimedia event detection and evidence recounting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2568–2577.

[47] J. Liu *et al.*, "Video event recognition using concept attributes," in *Proc. IEEE Workshop Appl. Comput. Vis.*, 2013, pp. 339–346.

[48] 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.

[49] Z. Gao *et al.*, "Er3: A unified framework for event retrieval, recognition and recounting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2253–2262.

[50] C. Gan, C. Sun, and R. Nevatia, "Deck: Discovering event composition knowledge from web images for zero-shot event detection and recounting in videos," in *Proc. Conf. Artif. Intell.*, 2017, pp. 4032–4038.

[51] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[53] S. Bell and K. Bala, "Learning visual similarity for product design with convolutional neural networks," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 98.

[54] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Deep convolutional matching," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 1164–1172.

[55] K. He, X. Zhang, S. Ran, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Eur. Conf. Comput. Vis.*, 2014, pp. 346–361.

[56] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 392–407.

[57] P. Sermanet *et al.*, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Int. Conf. Learn. Represent.*, Apr. 2014.

[58] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[59] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3286–3293.

[60] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[61] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.

[62] P. Over *et al.*, "TRECVID 2013: An overview of the goals, tasks, data evaluation mechanisms, and metrics," in *Proc. TREC Video Retrieval Eval.*, 2013.

[63] Y. Jia *et al.*, "CAFFE: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[64] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 1469–1472.
[65] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27–65, 2011.
[66] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[67] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.
[68] P. Over *et al.*, "TRECVID 2012—An overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proc. TREC Video Retrieval Eval.*, 2012.
[69] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 21–29.
[70] Y. Yan *et al.*, "Image classification by cross-media active learning with privileged information," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2494–2502, Dec. 2016.
[71] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *Int. J. Comput. Vis.*, vol. 113, no. 2, pp. 113–127, 2015.

Hao Zhang received the B.Sc. degree from Nanjing University, Nanjing, China, in 2012, the M.Sc. degree from the Chinese University of Hong Kong, Hong Kong, in 2013. He is currently working toward the Ph.D. degree in computer science at the City University of Hong Kong, Hong Kong.

He is currently with the VIREO Group, City University of Hong Kong. His research interest include multimedia content analysis, including semantical concept indexing and multimedia event detection.

Chong-Wah Ngo received the B.Sc. and M.Sc. degrees in computer engineering from Nanyang Technological University, Singapore, and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong.

He is currently a Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. Before joining the City University of Hong Kong, he was a Postdoctoral Scholar with the Beckman Institute, University of Illinois at Urbana-Champaign (UIUC), Urbanna, IL, USA. He was also a Visiting Researcher with Microsoft Research Asia, Beijing, China. His research interests include large-scale multimedia information retrieval, video computing, multimedia mining, and visualization.

He was the Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA (2011–2014). He was the Conference Co-Chair of the ACM International Conference on Multimedia Retrieval 2015 and the Pacific Rim Conference on Multimedia 2014. He also served as Program Co-Chair of ACM Multimedia Modeling 2012 and ICMR 2012. He was the Chairman of ACM (Hong Kong Chapter) from 2008 to 2009.