DevNet: A Deep Event Network for Multimedia Event Detection and Evidence Recounting

Chuang Gan[1]* Naiyan Wang[2]* Yi Yang[3] Dit-Yan Yeung[2] Alexander G. Hauptmann[4]
[1] Institute for Interdisciplinary Information Sciences, Tsinghua University, China
[2] Hong Kong University of Science and Technology
[3] Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Australia
[4] School of Computer Science, Carnegie Mellon University, USA
{ganchuang1990, winsty, yee.i.yang }@gmail.com,
dyyeung@cse.ust.hk, alex@cs.cmu.edu

Abstract

In this paper, we focus on complex event detection in internet videos while also providing the key evidences of the detection results. Convolutional Neural Networks (CNNs) have achieved promising performance in image classification and action recognition tasks. However, it remains an open problem how to use CNNs for video event detection and recounting, mainly due to the complexity and diversity of video events. In this work, we propose a flexible deep CNN infrastructure, namely Deep Event Network (DevNet), that simultaneously detects pre-defined events and provides key spatial-temporal evidences. Taking key frames of videos as input, we first detect the event of interest at the video level by aggregating the CNN features of the key frames. The pieces of evidences which recount the detection results, are also automatically localized, both temporally and spatially. The challenge is that we only have video level labels, while the key evidences usually take place at the frame levels. Based on the intrinsic property of CNNs, we first generate a spatial-temporal saliency map by back passing through DevNet, which then can be used to find the key frames which are most indicative to the event, as well as to localize the specific spatial position, usually an object, in the frame of the highly indicative area. Experiments on the large scale TRECVID 2014 MEDTest dataset demonstrate the promising performance of our method, both for event detection and evidence recounting.

1. Introduction

Detecting complex events in videos is a challenging task which has received significant research attention in the computer vision community. Compared to atomic concept recognition, which mainly focuses on recognizing particular objects and scene in still images or simple motions in short video clips of 5-10 seconds, multimedia event detection deals with more complex videos that consist of various interactions of human actions and objects in different scenes often lasting for several minutes to even an hour. Thus, an event is a semantic abstraction of video sequences of higher level than a concept and often consists of multiple concepts. For example, a *"Town hall meeting"* event can be described by multiple objects (e.g., *persons, podium*), a scene (e.g., *in a conference room*), actions (e.g., *talking, meeting*) and acoustic concepts (e.g., *speech, clapping*). Besides the concern of detecting semantic events, in many situations just assigning a video an event label is not enough, as discussed in [35, 34, 25, 45], because a long unconstrained video may contain a lot of irrelevant information and even the same event label may contain large intra-class variations. Besides providing a single event label, many users also

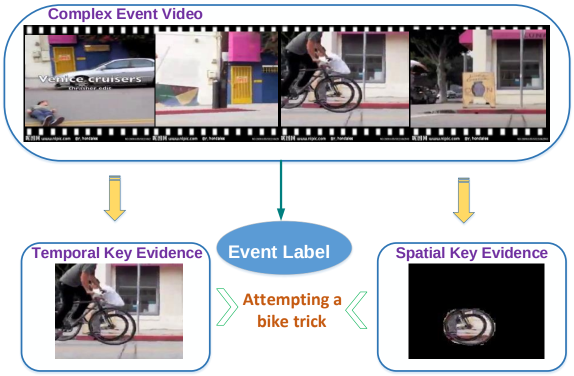

Figure 1. Given a video for testing, DevNet not only provides an event label but also spatial-temporal key evidences.

*The first two authors contribute equally to this work.

want to know why the video is recognized as this event, namely the key evidences that lead to the detection decision. This process is called Multimedia Event Recounting (MER). If event detection answers the question "Is this the desired event?", event recounting answers the next question, "Why does this video contain the desired event?". The key step of event recounting is localizing the key evidences, referred to here as *evidence recounting*, which is the focus of this paper. The recounting result is category-specific. It is unlike traditional video summarization tasks [24, 27, 12], which mainly seek to reduce the redundancy of the videos. As a result, the localized key evidences could effectively summarize the videos, and then allow the users to browse the videos and grasp the important parts quickly.

Although many algorithms have been proposed for the event detection and recounting problems recently, the challenges have not been fully addressed. The most successful methods for event detection are still aggregating shallow hand-crafted visual features, e.g., SIFT [26], MOSIFT [4], trajectory [37], improved dense trajectory [38], followed by feature pooling and a conventional classifier, such as Support Vector Machine (SVM) [2] or Kernel Ridge Regression (KR) [36]. However, such shallow features and event detection pipeline cannot capture the high complexity and variability of unconstrained videos. For recounting, most of the previous works focus on generating temporal-level key evidences (informative key frames or shots). However, this is still far from satisfactory because even within one frame, the scene and objects may be cluttered and non-informative. As the example in Figure 1 shows, the spatial localized *bike wheel* may suggest that this video is an *"Attempting a bike trick"* event, while the background *window* does not. Therefore, we decide to take it a step further, not only to localize the temporal evidences but also the spatial evidences. However, simultaneously assigning the retrieved video an event label and providing spatial-temporal key evidences is a non-trivial task due to the following reasons. First, different video sequences of the same event may have dramatic variations. Taking the *"Winning a race without vehicle"* event as an example, it may take place in a stadium, in a swimming pool or even in an urban park, where the visual features could be very different. Therefore, we can hardly utilize the rigid templates or rules to localize the key evidences. Second, the cost of collection and annotation of spatial-temporal key evidences is generally extremely high. It is prohibitive to extend the traditional fully-supervised object localization approaches for images, which employ the ground-truth bounding box information for training, to the video event recounting task directly.

In contrast to hand-crafted features, learning features with Convolutional Neural Networks (CNNs) [17], has shown great potentials in various computer vision tasks giving state-of-the-art performance in image recognition [17, 11, 3, 13, 40, 39] and promising results in action recognition [16, 32]. The successes of CNNs also shed light on the multimedia event detection and recounting problems. However, whether and how the CNN architecture could be exploited for the video event detection and recounting problems has never been studied before. This motivates us to apply CNNs to detecting and recounting event videos.

In this paper, we propose a Deep Event Network (DevNet) that can simultaneously detect high-level events and localize spatial-temporal key evidences. To reduce the influence of limited training data, we first pre-train the DevNet using the largest image dataset to date, ImageNet [7], and then transfer the image-level features and train a new video-level event detector by fine-tuning the network. Next, we exploit the intrinsic property of CNNs to generate a spatial-temporal saliency map without resorting to additional training steps. We only need to rank the saliency scores on the key frame level to localize the informative temporal evidences. For the top ranked key frames, we apply the graph-cut algorithm [1] to the segmentation of discriminative regions as the spatial key evidences. Note that the localization process only utilizes the video-level event label without requiring the annotations of key frames and bounding boxes. Our work makes the following contributions:

- To the best of our knowledge, we are the first to conduct high-level video event detection and spatial-temporal key evidence localization based on CNNs.
- This is the first paper that attempts to not only localize temporal key evidences (informative key frames and shots), but also provide discriminative spatial regions for evidence recounting.
- We show that our framework significantly outperforms state-of-the-art hand-crafted shallow features on event detection tasks and achieves satisfactory results for localizing spatial-temporal key evidences, which confirm the importance of representation learning for the event detection and evidence recounting tasks.

The rest of this paper is organized as follows. In Section 2, we review related work in multimedia event detection, multimedia event recounting and CNNs. Section 3 presents the DevNet and in particular details on how it can be applied to multimedia event detection and recounting. The experimental settings and evaluation results are presented in Section 4. Section 5 concludes the paper.

2. Related Work

Our framework relates to three research directions: event detection, event recounting, and CNNs, which will be briefly reviewed in this section.

2.1. Event Detection

Complex event detection has attracted a lot of research interest in the past decade. A recent review can be found in [15]. A video event detection system usually consists of the following procedure: feature extraction, quantization/pooling, and classifier training. Many event detection approaches rely on shallow low-level features such as SIFT [26] for static key frames, and STIP [20] and MOSIFT [4] for videos. Recently, state-of-the-art shallow video representation makes use of dense point trajectories [37, 38]. Its feature vectors are obtained by tracking densely sampled points and describing the volume around tracklets by histograms of optical flow (HOF) [21], histograms of oriented gradients (HOG) [5], and motion boundary histograms (MBH) [6]. To aggregate video-level features, it then applies Fisher vector coding [29] on the shallow low-level features. Moreover, there are also several attempts to conduct few-shots [44, 28] and even zero-shot [42, 10] event detection.

2.2. Event Recounting

Multimedia event recounting aims to find the event-specific discriminative parts of video. Most existing approaches focus on the temporal domain. In [34, 25, 45, 30], they apply object and action detectors or low-level visual features to localize temporal key evidences. They train a video-level classifier and then use it to rank the key frames or shots. These approaches are based on the assumption that the video-level classifiers that can distinguish positive and negative exemplars can also be used to distinguish the informative shots. However, these approaches equally treat the shots or key frames within the video. Consequently, the classifier may be confused by the ubiquitous but non-informative shots from videos. To overcome these limitations, [18] and [19] formulated the problem as a multiple-instance learning problem, aiming at learning an instance-level event detection and recounting model by selecting the informative shots or key fames during the training process. However, these approaches could only localize temporal key evidences.

2.3. Convolutional Neural Networks

Deep learning tries to model high-level abstraction of data by using model architectures composed of multiple nonlinear transformations. Specifically, CNNs [23] correspond to a biologically-inspired class of deep learning models that have demonstrated extraordinary abilities for some high-level vision tasks, such as image classification [17], object detection [11], and scene labeling [9]. Moreover, the features learned by large networks trained on the ImageNet dataset [7] show great generalization ability that yields state-of-the-art performance beyond standard image classification tasks, e.g., on several action recognition datasets [16, 32]. Besides, the problem of understanding and visualizing deep CNNs [47, 22, 8] has also attracted a lot of research attention. Very recently, [31, 47] proposed to localize the objects in images in a weakly supervised manner without relying on bounding box annotations. Compared to still image data and shot action videos, there is relatively little work on applying CNNs to multimedia event detection and recounting tasks. This motivates us to exploit the powerful features learned by CNNs to solve these problems.

3. DevNet Framework

In the proposed DevNet, the CNN architecture is similar to the network described in [17] except that it is much deeper. The CNN contains nine convolutional layers and three fully-connected layers. Between these two parts, a spatial pyramid pooling layer [13] is adopted. Consequently, without sufficient training data, it is very difficult to obtain an effective DevNet model for event detection. Thus, we first use the large ImageNet dataset [7] to pre-train the CNN for parameter initialization. The goal of this pre-training stage is to learn generic image-level features. However, directly using the parameters obtained from training on ImageNet for video event detection is not a proper choice, due to the domain difference between multimedia event detection and image classification. Thus we apply dataset-specific fine-tuning [3, 11, 41, 32] to adjust the parameters. After fine-tuning the parameters of the DevNet, we apply a single backward pass to identify the pixels in the same video with strong responses as spatial-temporal key evidences for event recounting. The DevNet framework is depicted in Figure 2.

3.1. DevNet Pre-training

Our experiments start with a deep CNN trained on the ILSVRC-2014 dataset [16] which includes 1.2M training images categorized into 1000 classes. The structure of our CNN is shown in Figure 2. It is implemented using the Caffe [14] toolbox. Given a training image, we first resize its shorter edge to 256 pixels. Then, we randomly extract fixed-size 224×224 patches from the resized images and train our network with these extracted patches. Each extracted patch is pre-processed by image mean subtraction, random illumination and contrast augmentation [33]. As described in [17], the output of the last fully-connected layer is fed into a 1000-way softmax layer with the multinomial logistic regression used to define the loss function, which is equivalent to defining a probability distribution over the 1000 classes. For all layers, we use Rectified Linear Units (ReLU) [17] as the nonlinear activation function. We train the network by using stochastic gradient descent with a momentum of 0.9 and weight decay of 0.0005. To

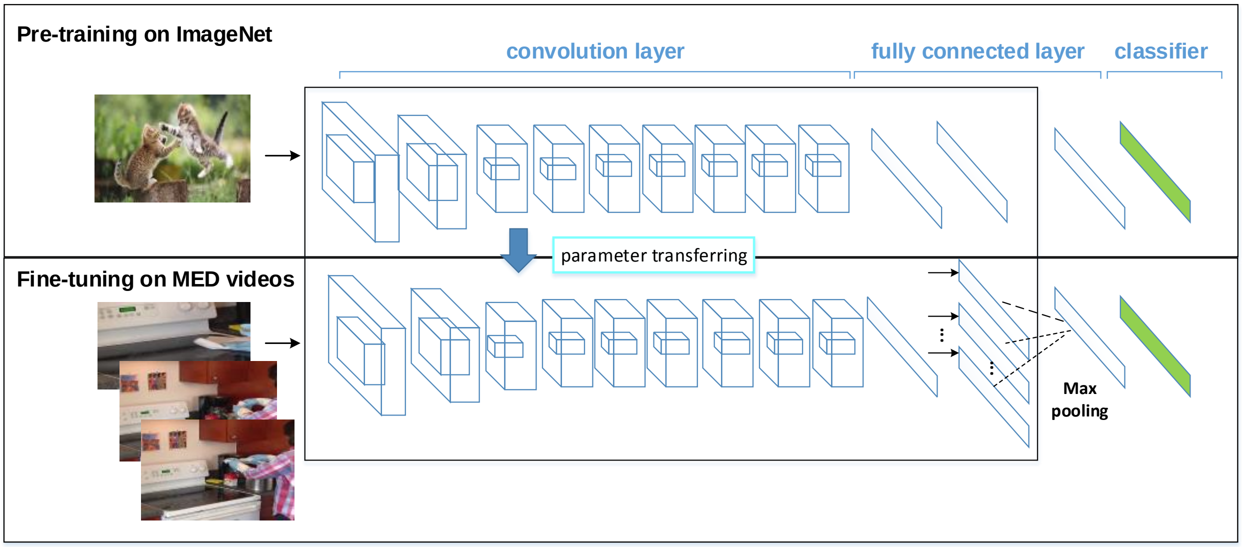

Figure 2. An illustration of the infrastructure of DevNet. We first pre-trained the DevNet using the ImageNet, and then fine-tuning on the MED video dataset.

overcome over-fitting, the first two fully-connected layers are followed by a dropout layer with a dropout rate of 0.5. The learning rate is initialized to 0.01 for all layers and is reduced to one-tenth of the current rate after every 20 epochs (80 epochs in all). Our weight layer configuration is: conv64-conv192-conv384-conv384-conv384-conv384-conv384-conv384-conv384-full4096-full4096-full1000. On ILSVRC2014 [7] validation set, the network achieves the top-1/top-5 classification error of 29.7%/10.5%.

3.2. DevNet Fine-tuning

Inspired by the recent success of domain-specific fine-tuning [11, 3, 41], we also utilize the training video data in the MED dataset as target data to further adjust the parameters. This is to adapt the model pre-trained on the ImageNet dataset to the video event detection task. The steps of video-level fine-tuning are described as follows.

First, we remove the softmax classifier and the last fully connected layer of the pre-trained network since it is specific to the ImageNet classification task. Next, to aggregate image-level features into the video-level representation, cross-image max-pooling is applied to fuse the outputs of the second fully-connected layer from all the key frames within the same video. Let $s_t = (s_{1t}, ..., s_{nt})^T \in \mathcal{R}^n$ be the feature vector of key frame t and $f = (f_1, ..., f_n)^T \in \mathcal{R}^n$ be the video-level feature vector. Then the i^{th} dimension of the video-level feature vector f can be represented as:

$$f_i = \max_t s_{it}. \tag{1}$$

Considering that the evaluation of event detection tasks is based on ranking, there is no inter-event competition and so we replace the softmax loss with a more appropriate c-way independent logistic regression, which produces a detection score for each event class. Here c denotes the number of event classes. To handle the imbalance of positive and negative exemplars, we randomly sample them in a 1:1 ratio during the fine-tuning process.

Implementation details. Similar to the approach described in [46], we sample several key frames from each video and then follow the same preprocessing and data augmentation steps as in the pre-training stage. Furthermore, as inspired by the experience of [11, 3], we use different learning rates for different layers. The learning rates of the convolutional layers, the first two fully-connected layers and the last fully-connected layer are initialized to 0.001, 0.001 and 0.01, respectively. We fine-tune for 30 epochs in total and decrease the learning rate to one-tenth of the current rate of each layer after 10 epochs.

After fine-tuning, we use the video-level representation after cross-image max-pooling (i.e., f in Eq. (1), the last fully connected layer) as the features for the event detection task.

3.3. Gradient-based Spatial-temporal Saliency Map

In this section, we extend a previous method [31] which generates class saliency maps for images to the video domain. The main idea of our event recounting approach is that, given a learned detection DevNet and a class of interest, we trace back to the original input image by a backward pass with which we can find how each pixel affects the final detection score for the specified event class.

Let us start with a motivational example. For a video V, we represent it as $X \in \mathcal{R}^{p \times q \times n}$, where p and q denote the height and width of each frame and n is the number of frames. We consider a simple case in which the detection score of event class c is linear with respect to the video, i.e.

$$S_c(V) = w_c^T x + b_c, \tag{2}$$

where $x \in \mathcal{R}^{pqn \times 1}$ is a vectorized form of the video V, and $w_c \in \mathcal{R}^{pqn \times 1}$ and b_c are the weight vector and bias of the model. In this case, it is easy to see that the magnitude of the elements of w_c specifies the importance of the corresponding pixels of V for class c.

In the case of a deep CNN, however, the class score $S_c(V)$ is a highly nonlinear function of V, so the assumption and analysis in the previous paragraph cannot be applied directly. However, we can approximate $S_c(V)$ by a first-order Taylor expansion expanding at V_0, where

$$S_c(V) \approx w_c^T x + b, \tag{3}$$

with the derivative of $S_c(V)$ with respect to V at point V_0 as:

$$w_c = \left.\frac{\partial S_c}{\partial V}\right|_{V_0}. \tag{4}$$

The magnitude of the derivative of Eq. (4) indicates which pixels within the video need to be changed the least to affect the class score the most. We can expect that such pixels are the spatial-temporal key evidences to detect this event.

Given a video that belongs to event class c with k key frames of size $p \times q$, the spatial and temporal key evidences are computed as follows. First, the derivative w_c in Eq. (4) is found by back-propagation. After that, the saliency map is obtained by rearranging the elements of vector w_c. In the case of a gray scale image, the number of elements in w_c is equal to the number of pixels in each frame multiplied by the number of key frames. So the saliency score of each pixel in each key frame can be computed as $M(i, j, k) = |w_c(h(i, j, k))|$, where $h(i, j, k)$ is the index of the element of w_c corresponding to the image pixel in the i^{th} row and j^{th} column of the k^{th} key frame. In the case of a multi-channel (*e.g.* RGB) image, we take the maximum magnitude of w_c across all color channels of each pixel as the saliency value. Thus for each event class, we can derive a single class-specific saliency score for each pixel in the video.

It is important to note that our spatial-temporal saliency maps $M \in \mathcal{R}^{p \times q \times n}$ are extracted using the DevNet trained on the video-level label and hence no additional annotation (such as informative key frames and bounding boxes) is required. The computation of saliency maps is extremely fast since it only requires a single backward pass without additional training.

After obtaining the spatial-temporal saliency map, we average the saliency scores of all the pixels within a key frame to obtain a key-frame level saliency score, and then we rank the key-frame level saliency scores to obtain the informative key frame. For the top ranked key frames, we use the saliency scores as guidance and apply the graph-cut algorithm [1] to segment the spatial salient regions.

4. Experiments

We present the dataset, experimental settings, evaluation criteria and experimental results in this section.

4.1. Evaluation Dataset

We perform our experiments on the challenging NIST TRECVID 2014 Multimedia Event Detection dataset *. To the best of our knowledge, it is the largest publicly available video corpora in the literature for event detection and recounting. This dataset contains unconstrained web videos with large variation in length, quality and resolution. In addition, it also comes with ground-truth video-level annotations for 20 event categories. Following the 100EX evaluation procedure outlined by the NIST TRECVID detection task, we used three different partitions for evaluation: *Background*, which contains about 5000 background videos not belonging to any of the target events, and *100EX*, which contains 100 positive videos for each event, are used as the training set. *MEDTest*, which contains 23,954 videos, is used as the test set.

4.2. Event Detection Protocol

The event detection task is to rank the videos in the database according to the specific query. We may also regard it as a video retrieval task. Our event detection approach consists of the following consecutive steps:

1. **Extracting key frames.** As processing all MED video frames will be computationally expensive, we only extract features from the key frames. Thus we start with detecting the shot boundaries by calculating the color histograms for all the frames. For each frame, we then subtract the previous color histogram from the current one. If the absolute value of the difference is larger than a certain threshold, this key frame is marked as a shot boundary [46]. After detecting the shot, we use the frame in the middle to represent that shot. By using this algorithm, we extracted about 1.2 million key frames from the TRECVID MED 2014 dataset.
2. **Extracting features.** We use the features of the last fully-connected layer after cross-frame max-pooling for video representation. We then normalize the features to make the l_2 norm of the feature vector equal

*http://nist.gov/itl/iad/mig/med14.cfm

to 1. More details are presented in Section 3.2.

3. **Training event classifier.** Due to limited training data in the video level, directly using the classifier in DevNet will result in inferior performance. Support vector machines (SVMs) [2] and kernel ridge regression (KR) [36] with χ^2 kernel are used. We obtain the regularization parameters by 5-fold cross validation.
4. **Testing event classifier**. We apply the trained event classifier on MEDTest and rank the videos by their detection results.

4.3. Event Detection Results

We use two evaluation metrics for ranked lists which are used by the NIST: Minimal Normalized Detection Cost (MinNDC) and Average Precision (AP) for each event. The definition of MinNDC is:

$$\text{MinNDC} = \frac{C_{MD} \times P_{MD} \times P_T + C_{FA} \times P_{FA} \times (1 - P_T)}{\min(C_{MD} \times P_T, C_{MD} \times (1 - P_T))}. \tag{5}$$

Here P_{MD} is the miss detection probability and P_{FA} is the false positive rate. $C_{MD} = 80$ is the cost for miss detection, $C_{FA} = 1$ is the cost for false alarm, and $P_T = 0.001$ is a constant which specifies the prior rate of event instances. Average Precision (AP) is a common metric for evaluation of ranking list. We also use the mean Average Precision (mAP) to evaluate the results by averaging all the events. A lower MinNDC or a higher AP and mAP value indicates better detection performance.

We compare our method with state-of-the-art hand-crafted features, improved dense trajectory with Fisher vector (IDTFV) for the event detection. We adopt the software of improved trajectories provided by Heng *et al.* [38] to extract raw trajectory features for each video in the MED14 dataset with default parameters, that is, frames of length 15 for each trajectory on a dense grid with 5-pixel spacing. We use PCA to reduce the dimensionality of the raw trajectory features from 426 to 213. Then we aggregate the features for each video using a Fisher vector [29] with 256 Gaussians, resulting in a 109,056-dimensional vector. We also follow the suggestions to apply power normalization and l_2 normalization to the feature vectors.

Table 1 reports experimental results making comparison with state-of-the-art shallow features with a single modality. From the results, we can see that the proposed CNN-based DevNet has 5.86% improvements in terms of mean Average Precision (mAP) compared with the state-of-the-art IDTFV shallow features by averaging over all events, which validates the effectiveness of the learned representation by DevNet approach.

4.4. Evidence Recounting Protocol

The goal of multimedia event evidence recounting is to give spatial-temporal key evidences for the videos detected as positive. Our event recounting approach consists of the following steps:

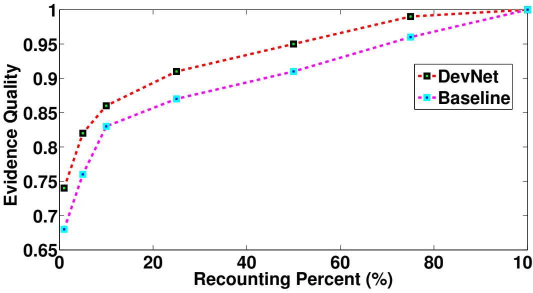

Figure 3. Comparison in terms of evidence quality against recounting percentage.

Table 2. Event recounting results comparing with the baseline approach. T means temporal key evidences and S means spatial key evidences.

ID	Baseline(T)	DevNet(T)	Baseline(S)	DevNet(S)
E021	3	**7**	1	**9**
E022	4	**6**	1	**9**
E023	1	**9**	0	**10**
E024	3	**7**	3	**7**
E025	5	5	3	**7**
E026	3	**7**	1	**9**
E027	**6**	4	0	**10**
E028	4	**6**	0	**10**
E029	5	5	5	5
E030	4	**6**	0	**10**
E031	5	5	0	**10**
E032	2	**8**	3	**7**
E033	3	**7**	0	**10**
E034	3	**7**	2	**8**
E035	4	**6**	3	**7**
E036	5	5	4	**6**
E037	2	**8**	2	**8**
E038	4	**6**	0	**10**
E039	3	**7**	4	**6**
E040	3	**7**	3	**7**
Average	3.6	**6.4**	1.75	**8.25**

1. **Extracting key frames**. It is the same as that for the event detection task.
2. **Spatial-temporal saliency map.** Given the event label we are interested in, we perform a backward pass based on the DevNet model to assign to each pixel in the testing video a saliency score. The higher score a pixel gets, the more likely it contributes to the key evidence. More details can be found in Section 3.3.
3. **Selecting informative key frames.** For each key frame, we compute the average of the saliency scores of all pixels and use it as the key-frame level saliency score. A higher score indicates that the key frame is more discriminative. We use the key frames with the N highest scores as temporal key evidences.
4. **Segmenting discriminative regions.** We use the spa-

Table 1. Event detection results comparing with improved dense trajectory Fisher vector (IDTFV). LOWER MinNDC / HIGHER AP indicates BETTER performance. The best results are highlighted in bold .

Event Description	ID	Evaluation Metric	IDTFV (SVM)	IDTFV (KR)	DevNet (SVM)	DevNet (KR)
Attempting a bike trick	E021	AP	0.1131	0.0986	**0.2887**	0.2741
		MinNDC	**0.1684**	0.2984	0.2413	0.2293
Cleaning an appliance	E022	AP	0.2406	0.2190	**0.2449**	0.1998
		MinNDC	0.553	0.5783	0.4270	**0.4251**
Dog show	E023	AP	0.6554	0.6609	0.7251	**0.7504**
		MinNDC	0.2695	0.1705	**0.0423**	0.0537
Giving directions to a location	E024	AP	0.0898	0.0542	**0.1059**	0.1022
		MinNDC	0.8294	0.8164	0.6532	**0.6510**
Marriage proposal	E025	AP	0.1187	**0.1349**	0.0782	0.0481
		MinNDC	0.8290	**0.8244**	0.8626	0.8725
Renovating a home	E026	AP	0.1400	0.1683	0.1880	**0.1911**
		MinNDC	0.7053	0.7026	0.5647	**0.5313**
Rock climbing	E027	AP	0.2258	0.2034	**0.2272**	0.2230
		MinNDC	0.3691	0.3931	0.3706	**0.3529**
Town hall meeting	E028	AP	0.4024	0.3474	**0.4286**	0.3831
		MinNDC	0.4180	0.4023	0.3805	**0.3554**
Winning a race without a vehicle	E029	AP	0.2162	0.2346	**0.2486**	0.2463
		MinNDC	0.3024	0.2946	**0.2637**	0.2658
Working on a metal crafts project	E030	AP	**0.2145**	0.1985	0.1606	0.2063
		MinNDC	0.6900	0.5881	0.5248	**0.5165**
Beekeeping	E031	AP	0.6313	0.6790	**0.8238**	0.8041
		MinNDC	0.2693	0.1935	0.0817	**0.0634**
Wedding shower	E032	AP	0.2140	0.1588	0.2716	**0.3149**
		MinNDC	0.4847	0.5335	**0.3563**	0.4465
Non-motorized vehicle repair	E033	AP	0.3489	0.3645	0.5787	**0.6354**
		MinNDC	0.4497	0.3708	**0.2306**	0.2316
Fixing musical instrument	E034	AP	0.2091	0.2727	0.4453	**0.4633**
		MinNDC	0.3458	0.3494	**0.2365**	0.2475
Horse riding competition	E035	AP	0.3526	0.3851	0.3943	**0.4409**
		MinNDC	0.3164	0.2768	**0.2606**	0.2820
Felling a tree	E036	AP	0.1947	**0.2611**	0.2271	0.1979
		MinNDC	0.4552	**0.3399**	0.4193	0.4331
Parking a vehicle	E037	AP	0.2633	0.2848	0.3337	**0.3735**
		MinNDC	0.5537	0.5337	0.1455	**0.1314**
Playing fetch	E038	AP	0.0676	0.0731	0.1093	**0.1115**
		MinNDC	0.5198	**0.4972**	0.5411	0.5584
Tailgating	E039	AP	**0.4648**	0.4529	0.4035	0.3929
		MinNDC	0.3354	0.3464	**0.3235**	0.3339
Tuning musical instrument	E040	AP	0.2283	0.2347	0.2937	**0.2982**
		MinNDC	0.4834	0.4768	0.4486	**0.4186**
Average		AP	0.2696	0.2743	0.3288	**0.3329**
		MinNDC	0.4674	0.4493	**0.3687**	0.3699

tial saliency maps of the selected key frames for initialization and apply graph-cut [1] to segment the discriminative regions as spatial key evidences.

For the temporal key evidence localization task, we compare our results with a state-of-the-art approach [35], which won the first place in the NIST TRECVID 2013 MER task. To the best of our knowledge, we are the first to deal with spatial key evidence localization and hence there exist no algorithms for comparison. Thus, we compare with the unsupervised salient object detection approach [48] in the selected key frames to generate spatial key evidences.

4.5. Evidence Recounting Result

Evaluation of video recounting results is difficult because no ground-truth information is available. Thus we conducted an experiment based on human evaluation. Two criteria were used: **evidence quality**, which measures how well the localized key evidences can convince the judge that a specific event occurs in the video; and **recounting percent**, which measures how compact the video snippets are compared to the whole video. A few volunteers were asked to serve as evaluators. Before evaluation, each evaluator was shown the event category descriptions in text as well as 10 positive examples in the training set. For each event, we used all the positive videos from MEDTest for evaluation.

During the evaluation process, the evaluators were first shown 1, 5, 10, 25, 50, 75 and 100 percents of the test videos separately. They voted on whether the key frames shown could convince them that it is a positive exemplar. We show the comparison results by plotting the evidence

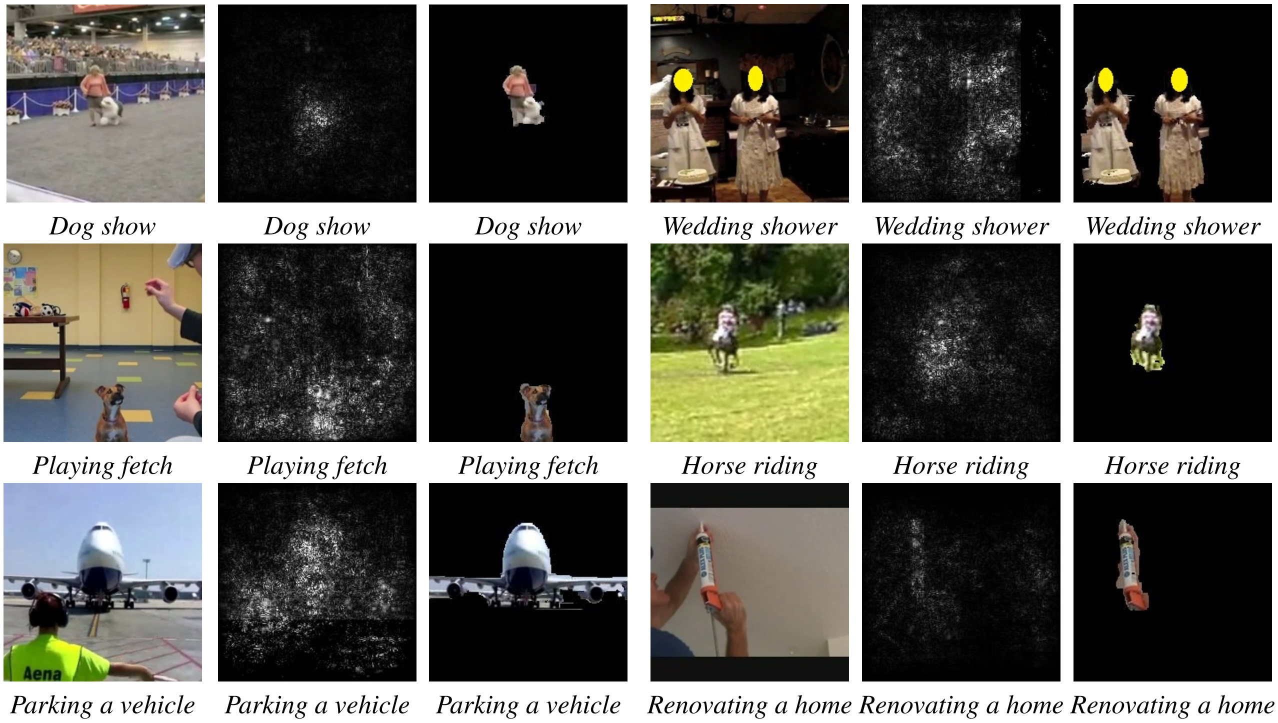

Figure 4. Event recounting results generated by DevNet. From left to right are top one temporal key evidence, spatial saliency map, and spatial key evidence.

quality (percentage of test videos convinced as positive exemplars) against the recounting percentage in Figure 3. Then the evaluators were presented the temporal key evidences (key frames) generated by [35] and DevNet with fixed recounting percentage (5%) and the spatial key evidences generated by [48] and DevNet. They voted on which key evidences are more informative. The voting results are shown in Table 2.

From Figure 3, we can see that DevNet can reduce the recounting percentage by 15% to 25% to get the same evidence quality as the baseline method. This validates that our approach provides reasonably good evidences for users to rapidly and accurately grasp the basic ideas of the video events. Table 2 summarizes the evaluators' preferences between our approach and the approach compared for each event. It can be seen that DevNet is better for most of the events. Some visual results are also shown in Figure 4.

5. Conclusion

In this paper, we presented a novel DevNet framework to address the video event detection and evidence recounting problems. Based on the proposed DevNet, the CNN pre-trained on large-scale image datasets, e.g, ImageNet, can be successfully transferred to the video domain. In addition, we apply a single back pass on DevNet (no additional annotations are required) to localize the spatial-temporal key evidences for the event recounting. We evaluate our experiment results on the challenging TRECVID MED 2014 dataset, and achieve a significant improvement than the state-of-the-art hand-crafted shallow features on the event detection task and satisfying event recounting results. We believe the event detection results could be further improved by the better model initialization and effective feature encoding [43]. In future work, we will add the motion information into the DevNet and also extend this method to generate tag descriptions for the spatial-temporal key evidences.

6. Acknowledgement

This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003, partially supported by research grant FSGRF14EG36, and partially supported by Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. We thanks for the generous donation of the GPUs by NVIDIA. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

[1] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, pages 105–112, 2001. 4322, 4325, 4327

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. 4322, 4326

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 4322, 4323, 4324

[4] M.-y. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos. 2009. 4322, 4323

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 4323

[6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, pages 428–441. 2006. 4323

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 4322, 4323, 4324

[8] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. *Dept. IRO, Université de Montréal, Tech. Rep*, 2009. 4323

[9] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, 2013. 4323

[10] C. Gan, M. Lin, Y. Yang, Y. Zhuang, and A. G. Hauptmann. Exploring semantic inter-class relationships (SIR) for zero-shot action recognition. In *AAAI*, 2015. 4323

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, pages 580–587, 2014. 4322, 4323, 4324

[12] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *ECCV*, 2014. 4322

[13] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pages 346–361. 2014. 4322, 4323

[14] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. *http://caffe. berkeleyvision. org*, 2013. 4323

[15] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah. High-level event recognition in unconstrained videos. *International Journal of Multimedia Information Retrieval*, 2(2):73–101, 2013. 4323

[16] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 4322, 4323

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 4322, 4323

[18] K.-T. Lai, D. Liu, M.-S. Chen, and S.-F. Chang. Recognizing complex events in videos by learning key static-dynamic evidences. In *ECCV*, pages 675–688, 2014. 4323

[19] K.-T. Lai, D. Liu, M.-S. Chen, and S.-F. Chang. Video event detection by inferring temporal instance labels. In *CVPR*, pages 2251–2258, 2014. 4323

[20] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005. 4323

[21] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008. 4323

[22] Q. V. Le. Building high-level features using large scale unsupervised learning. In *ICASSP*, pages 8595–8598, 2013. 4323

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4323

[24] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, pages 3–2, 2012. 4322

[25] J. Liu, Q. Yu, O. Javed, S. Ali, A. Tamrakar, A. Divakaran, H. Cheng, and H. Sawhney. Video event recognition using concept attributes. In *WACV*, pages 339–346, 2013. 4321, 4323

[26] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 4322, 4323

[27] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *CVPR*, pages 2714–2721, 2013. 4322

[28] Z. Ma, Y. Yang, N. Sebe, and A. G. Hauptmann. Knowledge adaptation with partially shared features for event detection using few exemplars. pages 1789–1802, 2014. 4323

[29] D. Oneata, J. Verbeek, C. Schmid, et al. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013. 4323, 4326

[30] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, pages 540–555. 2014. 4323

[31] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 4323, 4324

[32] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 4322, 4323

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 4323

[34] C. Sun, B. Burns, R. Nevatia, C. Snoek, B. Bolles, G. Myers, W. Wang, and E. Yeh. ISOMER: Informative segment observations for multimedia event recounting. In *ICMR*, page 241, 2014. 4321, 4323

[35] C. Sun and R. Nevatia. DISCOVER: Discovering important segments for classification of video events and recounting. In *CVPR*, pages 2569–257, 2014. 4321, 4327, 4328

[36] V. Vovk. Kernel ridge regression. In *Empirical Inference*, pages 105–116. 2013. 4322, 4326

[37] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011. 4322, 4323

[38] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 4322, 4323, 4326

[39] X. Wang, D. F. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. *CVPR*, 2015. 4322

[40] X. Wang, L. Zhang, L. Lin, Z. Liang, and W. Zuo. Deep joint task learning for generic object extraction. In *NIPS*, pages 523–531, 2014. 4322

[41] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. CNN: Single-label to multi-label. *arXiv preprint arXiv:1406.5726*, 2014. 4323, 4324

[42] S. Wu, S. Bondugula, F. Luisier, X. Zhuang, and P. Natarajan. Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *CVPR*, pages 2665–2672, 2014. 4323

[43] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. *CVPR*, 2015. 4328

[44] Y. Yang, Z. Ma, Z. Xu, S. Yan, and A. G. Hauptmann. How related exemplars help complex event detection in web videos? In *ICCV*, pages 2104–2111, 2013. 4323

[45] Q. Yu, J. Liu, H. Cheng, A. Divakaran, and H. Sawhney. Multimedia event recounting with concept based representation. In *ACM Multimedia*, pages 1073–1076, 2012. 4321, 4323

[46] S.-I. Yu, L. Jiang, Z. Mao, X. Chang, X. Du, C. Gan, Z. Lan, Z. Xu, X. Li, Y. Cai, et al. Informedia@ TRECVID 2014 MED and MER. 4324, 4325

[47] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. 2014. 4323

[48] W. Zhu, S. Liang, Y. Wei, and J. Sun. Saliency optimization from robust background detection. In *CVPR*, pages 2814–2821, 2014. 4327, 4328