This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

ER3: A Unified Framework for Event Retrieval, Recognition and Recounting

Zhanning Gao[1], Gang Hua[2], Dongqing Zhang[2], Nebojsa Jojic[2], Le Wang[1],
Jianru Xue[1], and Nanning Zheng[1]

[1]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]Microsoft Research

Abstract

We develop a unified framework for complex event retrieval, recognition and recounting. The framework is based on a compact video representation that exploits the temporal correlations in image features. Our feature alignment procedure identifies and removes the feature redundancies across frames and outputs an intermediate tensor representation we call video imprint. The video imprint is then fed into a reasoning network, whose attention mechanism parallels that of memory networks used in language modeling. The reasoning network simultaneously recognizes the event category and locates the key pieces of evidence for event recounting. In event retrieval tasks, we show that the compact video representation aggregated from the video imprint achieves significantly better retrieval accuracy compared with existing methods. We also set new state of the art results in event recognition tasks with an additional benefit: The latent structure in our reasoning network highlights the areas of the video imprint and can be directly used for event recounting. As video imprint maps back to locations in the video frames, the network allows not only the identification of key frames but also specific areas inside each frame which are most influential to the decision process.

1. Introduction

Analysis of event videos is a very challenging task. In contrast with action recognition which is usually based on video clips a few seconds long [4, 32], the event classification is performed on videos which often last for several minutes or even hours. These videos often capture multiple human actions and may contain a variety of different objects across various scenes. For example, a "birthday party" event may take place at home or in a restaurant, with multiple objects coming into focus, e.g. a birthday cake, and may include a variety of activities that span multiple frames, e.g. singing the birthday song, or blowing out candles.

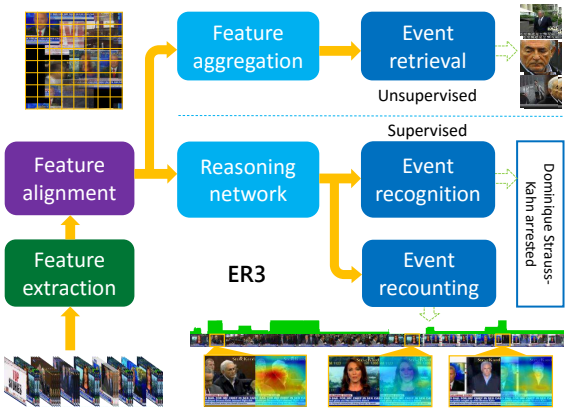

Figure 1. Illustration of ER3 framework for event retrieval, recognition and recounting. The compact video representation from feature aggregation can be used for large-scale event retrieval. With supervised training, ER3 can also recognize the event category of the input video. Event recounting falls directly out of the latent structure of the model in form of statistics displayed as heat maps for each frame indicating key areas related to the event.

In last decade, analysis of complex events in videos has attracted significant attention in the computer vision community [10, 11, 17, 24, 29, 35]. Previous research was pursued in both unsupervised and supervised settings. Unsupervised models were typically used for *event retrieval* [9, 29] where the goal is to retrieve all the related videos in the database in some sense similar to the query video provided by a user. On the other hand, supervised learning has been used in *event recognition* [3, 5] or detection [24, 47] in similar ways as in action recognition [4, 32] and general video classification [18, 46, 49]. In this latter case, a classifier is learned from annotated training videos to detect and recognize the event categories of the test videos, *e.g.*, the multimedia event detection task of the TRECVID [26]. In practical applications, it is often important to qualify the event category prediction by providing an explanation for it.

In particular the system needs to localize the key pieces of evidence that lead to the recognition decision. This is some times referred to as *event recounting*.

One of the key issues in event video analysis is the construction of appropriate video representations. For *event retrieval* and *event recognition*, the representation should be discriminative yet compact so that it can efficiently disambiguate various events in videos. Usually, a global feature vector is constructed based on frame-level appearance features for each event video [9, 10, 29, 46, 47]. In recognition tasks, this global video representation is then fed into either a linear classifier [47] or a neural network [18, 46] to identify the event category. However, this representation makes event recounting difficult, as tracing the decision back to individual image locations is intractable. Thus most existing systems perform event recounting as a post-processing step after recognition [10, 22, 40].

In this paper we propose a unified framework, named ER3, for *event retrieval, recognition* and *recounting*. Figure 1 illustrates the framework and the input/output of our ER3 system. In ER3, (i) we introduce a feature alignment step which can significantly suppress the redundant information and generate a more comprehensive and compact video representation called *video imprint*. In addition, the video imprint also preserves the local spatial layout among video frames. (ii) Based on the video imprint, we further employ a reasoning network, a modified version of the neural memory network [34], which can simultaneously recognize the event category and locate the key pieces of evidence for the event category. In fact, the recounting is so naturally integrated in the framework that the experiments show that the recounting step can assist the recognition task and improve the recognition accuracy. (iii) In the recounting task, not only do we predict the important frames related to the event as was done previously [22, 40], but we also jointly predict the important areas inside each frame since the local spatial layout is preserved in the video imprint.

The paper is organized as follows. Section 2 discusses related work about event videos analysis. Then, we present the technical details of the ER3 in Section 3. Experimental results are provided in Section 4. Finally, we conclude the paper in Section 5.

2. Related work

In unsupervised event retrieval, the goal is to retrieve all the related videos in the database associated with the query video. The key problem is to construct compact video representations. Previous methods [9, 29] often start building the video representation at the frame level. First, the local features, such as SIFT [23], are extracted from each frame and aggregated together to form a frame-level feature description based on encoding methods such as Fisher Vector [30, 28] or VLAD [16, 9]. Then, to form a video level representation. the frame-level descriptors are simply averaged across the video. Such sum-aggregation neglects the strong temporal correlations among consecutive frames. This may undesirably over-weight the information in certain long or recurrent shots in the video. We discuss this problem in Section 3 and show that the redundant information among frames can be effectively suppressed by the feature alignment step.

Event recognition or detection has attracted wide attention in the last decade. In general, event video recognition system can be divided into three stages: feature extraction, feature aggregation/pooling, training/recognition. As in event retrieval, the first two stages aim to build discriminative video representations. Previous work focused mostly on designing better video features or representations for the classifier, such as hand-crafted visual features [8, 23], motion features [41, 42], audio features [2], and mid-level concept/attributes features [7, 39]. Recently, the development of deep convolutional neural networks [20, 33] lead to promising results in event recognition tasks [18, 47, 50]. The video representations are usually constructed by directly aggregating the frame-level CNN features. These features vectors are typically used in conventional classifiers such as Support Vector Machine (SVM) [6], because of limited training data. In addition, several work [18, 50, 46, 51] also explore multiple features fusion strategies to further improve the recognition performance.

Event recounting refers to localization of key pieces of evidence in support of the recognition decision, a challenging task as only video-level annotations are provided. Event recounting is usually a post-processing step, performed after the recognition [22, 40]. Sun *et al.* [36] introduce an evidence localization model learned via a max-margin framework, and Chang *et al.* [7] employ a joint optimization framework with mid-level semantic concept representation for event recognition and recounting. Lai *et al.* [21] apply Multiple-Instance Learning (MIL) which can infer temporal instance labels as well as the video-level labels, since the videos are treated as sets of shots or instances. These recounting procedures only reason through time and usually at a coarse level. Gan *et al.* [10] train a deep event network for event recognition. In addition, it can also predict the key frames and the important areas inside the frames related to the event by backward passing the classification scores. This is still a post-processing method which does not assist in recognition, but rather attempts to explain it.

In contrast with these methods, at a core of our system is a generative model in which latent structure directly serves as a set of pointers to image locations. The model is trained in unsupervised way by jointly aligning the areas of different frames and estimating a distribution over features in corresponding areas. The resulting representation is a grid of distributions over features to which the frames are mapped,

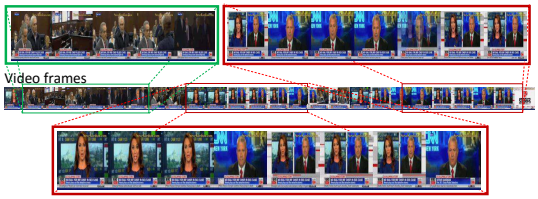

Figure 2. Illustration of the frames related to "Dominique Strauss-Kahn arrested". The frames in green box denote the positive frames related to the event. Red boxes show irrelevance frames.

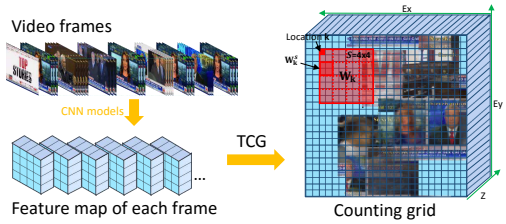

Figure 3. Illustration of tessellated counting grid (TCG). The right tensor block represents the counting grid with $\mathbf{E} = 24 \times 24$, $\mathbf{W} = 8 \times 8$, $\mathbf{S} = 4 \times 4$. Similar frames are usually represented in the same or nearby windows, *e.g.*, the anchor who we frequently see in the video.

much in a way image frames are mapped to panoramas in pixel space (see the toy example in Figure 1). This grid, along with the pointers back to the locations in the original frames forms a *video imprint* allowing us to consider image evidence in flexible ways. For example, the aggregation can be performed to emphasize mere presence rather than frequency of repetition of certain object or scene parts.

Our experiments show that video imprint aggregation yields better performance both in supervised and unsupervised cases compared with previously published work. The video imprint also allows us to reason over the local spatial layout of features found across frames. In order to predict the event class, our reasoning network analyzes the evidence present in different spatial locations of the compact video imprint much the way the attention mechanism in Memory Network [34] reasons over sentences in priming text. Recent work [48] also explored similar idea at video face recognition. In the process, the reasoning network highlights the areas of the imprint, which in turn map back to video frames and their corresponding spatial locations. In this way, recounting is an integral part of decision making, not merely a post processing step.

3. The details of ER3

In this section, we present the details of each module shown in Figure 1 and demonstrate how this framework perform event retrieval, recognition and recounting tasks.

3.1. Feature extraction

Recently, image descriptors based on the activations within deep convolutional neural networks (CNN) have emerged as state of the art generic descriptors for visual recognition [13, 31, 50]. Different from previous event video analysis methods [18, 50] which usually extract fully connected layers as the frame-level descriptors, we chose the activations of the last convolutional layers as the frame-level representation. Since the convolutional layer contains the spatial information of the input frame, we can perform more accurate recounting results beyond the frame level.

3.2. Feature alignment

Usually, the video representation is directly aggregated from the frame descriptors [9, 18, 47, 50]. However, this may undesirably over-weight the information in certain long or recurrent shots in the video, which can dominate the final representation. For instance, as shown in Figure 2, the shots of the anchor dominate in the event video related to the event "Dominique Strauss-Kahn arrested". Since those frames share similar content, simply averaging frame descriptors may lead to over-emphasis of these descriptors and reduce the discriminative power of the video representation. To mitigate this problem, we use feature alignment to balance the influence of frame features after feature extraction.

The idea of feature alignment comes from panoramic stitching [37, 38], which can stitch images into a full view panorama, removing the overlap among the input images. If we could generate an equivalent panoramic representation from video frames of an event, the redundancy across frames would be removed and the video representation would be less sensitive to the frequency of the repetition of the less discriminating features.

Obviously, the dynamic and complex event videos, are not a convenient target for frame stitching. Alignment at pixel level is difficult, and the frames are not mappable to a single panorama at any rate. To deal with geometric variation in objects or entire scenes in video frames, we first featurize the images using the activations of last convolutional layer extracted from each frame. Then, we employ the tessellated counting grid (TCG) model [27] to train a panorama in a generalized sense. The resulting grid of feature distributions contains multiple panoramas to which frames from different shots are automatically mapped. Thus the model captures spatial interdependence of the convolutional layer features in related frames, and also serves as the clustering grounds for different shots. The following is a brief introduction of TCG, please refer to [27] for detailed description.

Tessellated counting grid (TCG) [27] is designed to capture the spatial interdependence among image features. Given a set of images or a video sequence, it assumes that

each image/frame is represented by a set of l_1-normalized non-negative feature vectors $\{\mathbf{c}^s\}_{s\in\mathbf{S}}$ plugged in a tessellation $\mathbf{S} = S_x \times S_y$ [1]. Formally, the counting grid $\pi_{\mathbf{i},z}$ is a set of normalized features indexed by z (dimension of image feature) on the 2D discrete grid $\mathbf{i} = (i_x, i_y) \in \mathbf{E} = E_x \times E_y$, where $\mathbf{i}$ is the location on the grid.

As a generative model, the probability of generating the image features $\{\mathbf{c}^s\}_{s\in\mathbf{S}}$ from the window $W_\mathbf{k}$ in the location $\mathbf{k}$ of the grid is

$$p(\{\mathbf{c}^s\}_{s\in\mathbf{S}}|l=\mathbf{k}) = \mu \prod_z \prod_s \left(\sum_{\mathbf{i}\in\mathbf{W}^s_\mathbf{k}} \pi_{\mathbf{i},z} \right)^{c^s_z}, \quad (1)$$

where μ is the normalization constant. Thus, the joint distribution over the set of image features $\{\mathbf{c}^{s,t}\}_{s\in\mathbf{S},t\in T}$, indexed by t, and their corresponding latent window locations $\{l^t\}$ in the grid can be derived as

$$P(\{\mathbf{c}^{s,t}\},\{l^t\}) \propto \prod_t \sum_\mathbf{k} \prod_z \prod_s \left(\sum_{\mathbf{i}\in\mathbf{W}^s_\mathbf{k}} \pi_{\mathbf{i},z} \right)^{c^{s,t}_z}. \quad (2)$$

The counting grid π is estimated by maximizing the log likelihood of the joint distribution with an EM algorithm,

$$\begin{aligned} \text{E step}: \; & q(l^t=\mathbf{k}) \propto \exp\left(\sum_s \sum_z c^{s,t}_z \log \sum_{\mathbf{i}\in\mathbf{W}^s_\mathbf{k}} \pi_{\mathbf{i},z} \right), \\ \text{M step}: \; & \pi_{\mathbf{i},z} \propto \pi^{old}_{\mathbf{i},z} \sum_t \sum_s c^{s,t}_z \sum_{\mathbf{k}|\mathbf{i}\in\mathbf{W}^s_\mathbf{k}} \frac{q(l^t=\mathbf{k})}{\sum_{\mathbf{i}\in\mathbf{W}^s_\mathbf{k}} \pi^{old}_{\mathbf{i},z}}, \end{aligned} \quad (3)$$

where $q(l^t = \mathbf{k})$ denotes the posterior probability $p(l^t = \mathbf{k}|\{\mathbf{c}^{s,t}\}_{s\in\mathbf{S}})$ and $\pi^{old}_{\mathbf{i},z}$ is the counting grid at the previous iteration.

The iterative process of TCG will jointly estimate the counting grid π and align all training frame features to it. Thus, π summarizes the entire video, serving as a *video imprint*: Each of its locations corresponds to equivalent regions in a variety of frames, with this correspondence captured in q distribution above. See Figure 3 [2] for illustration.

3.3. Feature aggregation

In this section, we will demonstrate how to aggregate the video imprint into a compact video representation for unsupervised event retrieval. We refer each $\pi_\mathbf{i}$ on the video imprint as a counting grid descriptor. As shown in Figure 3, some counting descriptors are meaningless since no frames

[1] With l_1-normalization and appropriate down-sampling, the feature maps (after ReLU) from convolutional layer of CNN model naturally satisfy this assumption.

[2] We cannot directly visualize the counting grid of the frame features. For ease of illustration, we accumulate the frames on the location with the maximum posterior probability $q(l^t = \mathbf{k})$ and draw the mean image.

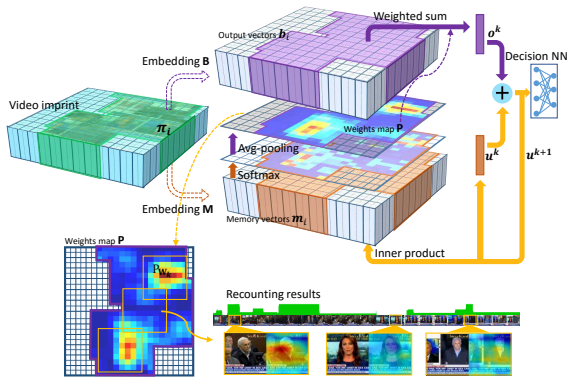

Figure 4. Illustration of reasoning network for event recognition and event recounting.

are aligned to their locations. The first step is to generate an active map for the video imprint to filter out the noisy counting grid descriptors associated with the location that aligned with few or no frames. Formally, the binary active map, $\mathbf{A} = \{a_\mathbf{i}|\mathbf{i}\in\mathbf{E}\}, a_\mathbf{i} \in \{0,1\}$, is computed as

$$a_\mathbf{i} = \begin{cases} 1 & \left\{ \mathbf{i}\in\mathbf{W}_\mathbf{k} | \; \mathbf{k} : \sum_{t=1}^{N} q(l_t=\mathbf{k}) > \tau \right\}, \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where τ is the threshold of the active map.

After generating the active map, the second step is quite simple: We apply sum-aggregation over the whole activated counting grid descriptors to produce the final video representation. Formally, the aggregation step can be written as

$$\phi_{FA}(\pi,\mathbf{A}) = \sum_{\mathbf{i}\in\mathbf{E}} a_\mathbf{i}\pi_\mathbf{i}. \quad (5)$$

The obtained $\phi_{FA}(\pi,\mathbf{A})$ is subsequently l_2-normalized and the cosine similarity is computed for event retrieval.

3.4. Reasoning over the imprint

Once the imprint is computed for a video, instead of reasoning over individual frame features, we can now reason in this compact representation in which each location corresponds to a recurring scene/object part, with the spatial layout of these locations mirroring the spatial layout of parts in the frames where they were seen. We treat locations in the imprint in a similar way the sentences are treated in a memory network [43, 34]. Our reasoning network makes the decision regarding the event category in stages which turn attention from one set of imprint location to the next (Figure 4). In the process, the imprint locations of importance are highlighted, and we can trace these highlights back to the locations in images that mapped to these areas of the imprint, as discussed above, using the q distributions and the

fact that the spatial layout of nearby locations in the imprint matches the layout in the original frames.

Our reasoning network differs from the memory networks in two ways. First, since there is no query question for event recognition, we initialize the input vector u^1 with Equation 5, *i.e.*, sum-aggregation of video imprint. Second, because the spatial organization in the imprint is meaningful, we add an average spatial pooling layer after the softmax layer in the original memory network architecture. The experiments show that, with the average pooling layer added, we can obtain smoother and more reasonable recounting results. The model details are as follows.

Memory layers in the reasoning network. As shown in Figure 4, the video imprint (the non-activated locations are ignored during the whole procedure) is processed via multiple memory layers (hops). In each layer, the counting grid descriptors $\pi_{\mathbf{i}}$ from video imprint are first embedded to output vector space and memory vector space with embedding matrices $\mathbf{B}$ and $\mathbf{M}$, respectively.

$$b_{\mathbf{i}} = \mathbf{B}\pi_{\mathbf{i}}, \;\; m_{\mathbf{i}} = \mathbf{M}\pi_{\mathbf{i}}, \tag{6}$$

where $b_{\mathbf{i}}$ denotes the output vector and $m_{\mathbf{i}}$ denotes memory vector. The memory vector $m_{\mathbf{i}}$ is used to compute the weights map $\mathbf{P} = \{p_{\mathbf{i}} | \mathbf{i} \in \mathbf{E}\}$ with the internal state u.

$$p_{\mathbf{i}} = \text{avgpooling}\left(\text{softmax}\left(u^T m_{\mathbf{i}}\right)\right). \tag{7}$$

The average pooling is performed with 3×3 windows, stride 1. The output vector o is then computed by a weighted sum over the output vectors $b_{\mathbf{i}}$.

$$o = \sum_{\mathbf{i}} p_{\mathbf{i}} b_{\mathbf{i}}. \tag{8}$$

For the internal state vector u, the initial u^1 computed with Equation 5, and the u^{k+1} in $k+1$ layer is computed as

$$u^{k+1} = u^k + o^k. \tag{9}$$

The final output vector is then fed into a decision network to predict the event category. It can be a simple softmax layer or have multiple fully connect layers. The recounting map of each frame shown in Figure 4 is generated via the sum of all weights maps, $\mathbf{P}^{\text{sum}} = \sum_k \mathbf{P}^k$, and the posterior probabilities $q(l^t = \mathbf{i})$ (More complex recounting inferences can also be made by showing conditional heat maps based on individual memory layers as we trace the reasoning engine through the layers). We use $\mathbf{P}^{\text{sum}}_{\mathbf{W}_{\mathbf{i}}}$ denotes the weights map cropped from $\mathbf{P}^{\text{sum}}$ in the window $\mathbf{W}_{\mathbf{i}}$. Then the recounting map $\mathbf{R}^t$ of frame t is

$$\mathbf{R}^t = \sum_{\mathbf{i}\in\mathbf{E}} q(l^t = \mathbf{i}) \mathbf{P}^{\text{sum}}_{\mathbf{W}_{\mathbf{i}}}. \tag{10}$$

The importance score of each frame is obtained with the sum of the recounting map.

4. Experiments

4.1. Datasets and evaluation protocol

In terms of event retrieval, we evaluated our method on a large-scale benchmark EVVE dataset [29]. It contains $2,995$ videos (620 videos are set as queries) related to 13 specific event classes. Given a single video of an event, the task is to retrieve videos related to the same event from the dataset. The methods are evaluated based on the mean AP (mAP) computed per event. The overall performance is evaluated by averaging the mAPs over the 13 events. In addition, a large distractors dataset ($100,000$ vedios) is also provided to evaluate the retrieval performance on large-scale data.

To evalaute the event recognition and recounting, we used three datasets: EVVE, Columbia Consumer Videos (CCV) [19] and TRECVID MEDTest 14 (MED14) [26].

In addition to using it in the event retrieval evaluation, we also configured the EVVE as a small recognition dataset. As such, it contains 13 events. For each event, we set the query video as the test data (620 videos), and treat the groundtruth in the dataset as the training data. We report the top-1 classification accuracy to evaluate the recognition performance.

The CCV dataset [19] contains $9,317$ YouTube videos belonging to 20 classes. We follow the protocol defined in [19] to use a traing set of $4,659$ and a test set of $4,658$ videos. The TRECVID MEDTest 14 [26] is one of the most challenging datasets for event recognition containing 20 complex events. In the training section, there are 100 positive exemplars per event, and all events share negative exemplars with about $5,000$ videos. The test data has approximately $23,000$ videos.

For these two datasets, mAP is used to evaluate the performance of event recognition according to the NIST standard [26]. Since there is no ground truth information for the recounting task, we only provide qualitative analysis for event recounting results.

4.2. Implementation details

Frame-level descriptor. Given an input video, we sample 5 frames per second (5 fps) to extract the CNN features. We exlpore various pre-trained CNN models, *i.e.*, AlexNet [20], VGG [33] and ResNet-50 [14] to evaluate our method. We adopt the output from the last convolutional layer (after ReLU) of these models as the frame descriptors. The CNN feature maps are down-sampled to 4×4 with linear interpolation to fit the TCG (we set $\mathbf{S} = 4 \times 4$ in TCG for computation efficiency). In addition, we also average all the frame descriptors over the video (sum-aggregation), as the baseline to evaluate our framework.

Post-processing. For the baseline video representation, we apply the same post-processing strategy as in [1,

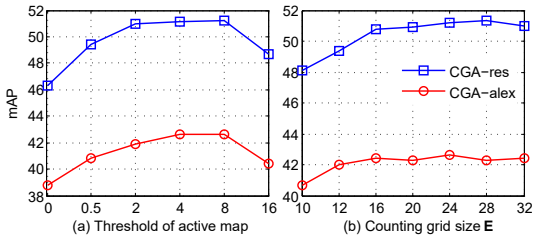

Figure 5. (a) The influence of the threshold τ of active map. (b) Counting grid aggregation with different counting grid sizes $\mathbf{E}$.

12], *i.e.*, the representation vector of a video is first l_2-normalized, and then whitened using PCA [15] and l_2-normalized again. For the counting grid descriptors on the video imprint, power normalization ($\alpha = 0.2$) shows better results than l_2-normalization in our experiments. Therefore, after feature alignment, the counting grid descriptors are first power normalized, then PCA-whitened and l_2-normalized.

Re-ranking methods for event retrieval. For the event retrieval task on the EVVE dataset, we also employ two variants of query expansion methods presented by Douze *et al.* [9]: Average Query Expansion (AQE) and Difference of Neighborhood (DoN). In our experiments, we set $N_1 = 10$ for AQE and $N_1 = 10, N_2 = 2000$ for DoN.

Training details for the reasoning network. The reasoning network (RNet) was trained with stochastic gradient descent (SGD) method. The initial learning rate was $\beta = 0.025$, which then annealed every 5 epochs by $\beta/2$ until 20 epoches were finished. All weights were initialized randomly from a Gaussian distribution with zero mean and $\sigma = 0.05$. The weights were shared among different memory layers. The batch size was 128 and the gradients with an l_2 norm larger than 20 were rescaled to norm 20 during the training step.

Computational complexity. The most time consuming step is constructing video imprint for input video. As discussed in [27], with efficient use of cumulative sums, the computational complexity of learning CG with EM algorithm grows at most linearly with the product of counting grid size and video length. In our experiments, the average running time of TCG (with ResNet features) for EVVE (about 1200 frames per video) implemented on the GPU platform (K40 with MATLAB parallel computing toolbox) is about 15 seconds.

4.3. Evaluation results on event retrieval

4.3.1 Parameter analysis

Threshold of the active map. Figure 5(a) shows the retrieval performance with different thresholds used for the active map construction. We can observe that increasing τ helps filter out some very short shots (with only several frames) which are usually not that meaningful. We set $\tau = 8$ in the subsequent experiments.

Representation	Dim.	mAP
Sum-alex	256	38.3
Sum-res	1024	46.6
Sum-(alex+res)	1280	47.3
CGA-alex	256	42.6
CGA-res	1024	51.2
CGA-(alex+res)	1280	**52.3**

Table 1. Comparison with sum-aggregation on EVVE dataset. Sum- and CGA- denote sum-aggregation and counting grid aggregation, respectively. alex and res denote two CNN models, AlexNet and ResNet-50. For ResNet based representation, the vectors dimension are reduced to 1024 with PCA-whitening. (alex+res) denotes the concatenated vector.

Counting grid size. To evaluate the influence of counting grid size, we first fix the window size ($\mathbf{W} = 8 \times 8$) and tessellation size ($\mathbf{S} = 4 \times 4$) of the counting grid. Then we chose 7 different counting grid size to perform the feature alignment. The performance for each size is presented in Figure 5(b). No further improvement can be obtained when $\mathbf{E} > 24$. Therefore, the size of counting grid is fixed to 24 for the following experiments.

4.3.2 Comparison with sum-aggregation

We refer to our unsupervised flow (combining the feature alignment and aggregation steps) on ER3 as counting grid aggregation (CGA). Table 4.3.2 shows the retrieval performance compared with baseline. We evaluate the CGA on two different CNN models, AlexNet [20] and ResNet-50 [14]. Our aggregation method obtains better retrieval performance, $\mathrm{mAP} = 52.3$, with the benefits from feature alignment step that can suppress the redundancy among frames. In addition, consistent improvement can be observed for different CNN models, *i.e.*, 11.2% gain with AlexNet and 9.9% with ResNet-50.

4.3.3 Comparison with state of the arts

In Table 4.3.3, we can see that the sum-aggregation with CNN features already achieves better results compared with previous work [29, 9]. After merging with 100K distractors, the mAP of CGA-(alex+res) achieves 42.9 which is also better than the baseline ($\mathrm{mAP} = 38.7$) and Hyperpooling [9] ($\mathrm{mAP} = 26.5$). In addition, the query expansion can further boost the performance. We achieve 36.6% and 8.9% improvement compared with previous result ($\mathrm{mAP} = 44.0$) and the baseline ($\mathrm{mAP} = 55.2$) on EVVE, respectively. Consistent improvement is also observed with query expansion on the large dataset (EVVE+100K).

Method	Dim.	EVVE			EVVE+100K		
			AQE	DoN		AQE	DoN
MMV [29]	512	33.4	–	–	22.0	–	–
CTE [29]	–	35.2	–	–	20.2	–	–
MMV+CTE	–	37.6	–	–	25.4	–	–
SSC [9]	16384	36.3	38.9	44.0	26.5	30.1	33.1
Sum-(alex+res)	1280	47.3	53.1	55.2	38.7	45.8	47.1
CGA-(alex+res)	1280	**52.3**	**58.5**	**60.1**	**42.9**	**50.4**	**52.7**

Table 2. Retrieval performance compared with other methods. AQE and DoN denote the two Re-ranking methods.

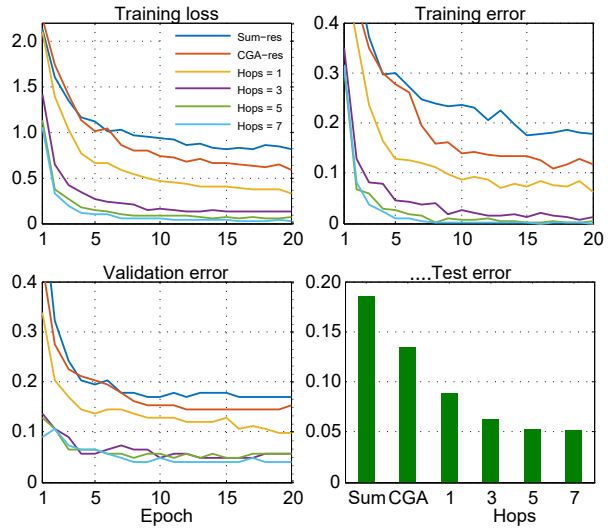

Figure 6. The influence of RNet with increased hops on EVVE dataset (Best viewed in color).

4.4. Evaluation results on event recognition

4.4.1 Parameter analysis

Structure of the reasoning network. For the EVVE dataset, we set a softmax layer as the decision network. The video imprint is generated based on the ResNet-50 [14] model and its counting grid descriptors are first reduced to 256 dimension with PCA-whitening before feeding to reasoning network (RNet). For CCV and MED14 datasets, we add a fully connected layer in front of the softmax layer as the decision network for better performance. Besides the ResNet-50 model, we also evaluate the framework with VGG (16 layers) [33] model for these two datasets. The dimension of the counting grid descriptors is set to 1024 and 512 for ResNet-50 and VGG, respectively. For all the datasets, the internal vectors b_i and m_i have the same dimension with the input counting grid descriptors.

Number of memory layers. Figure 6 illustrates the influence of RNet with increased hops on the EVVE dataset.

	Method	vgg	res	(vgg+res)
CCV	Sum-	74.3	75.3	78.1
	CGA-	75.7	76.6	79.1
	RNet-	76.7	78.5	**79.9**
MED14	Sum-	26.0	30.4	32.8
	CGA-	30.5	32.2	33.7
	RNet-	32.8	34.2	**36.9**

Table 3. Comparison with sum-aggregation and CGA. Sum- and CGA- denote sum-aggregation and counting grid aggregation, respectively. RNet- denote the reasoning network. vgg and res denote two CNN model, VGG and ResNet-50. (vgg+res) denotes the later fusion result.

To be fair comparison, we employ the same decision network as classifier for the baselines and the output from RNet. We compare with two representations, the video representations with sum-aggregation and counting grid aggregation (CGA). In fact, if we fix the value of the weights map equal to the active map, the RNet will reduce to the CGA, *i.e.*, the unsupervised flow in Figure 1. We can see that CGA provides better performance than sum-aggregation and the RNet can further refine the video representation and leads to better recognition accuracy than the two baselines. In addition, the gain is also increased with more hops. Consistent gains are observed on both CCV and MED14 datasets. We set the hops $= 3$ in the following experiments for CCV and MED14 datasets.

4.4.2 Performance on CCV and MED14

Table 4.4.2 shows the recognition performance (mAP) of RNet and baseline methods. With the benefit from reweighting the video imprint, the RNet achieves better results on the CCV (mAP = 79.9) and MED14 (mAP = 36.9) datasets compared with sum-aggregation and CGA. In addition, on the CCV dataset, we also employ the same strategy as [45] to combine motion and audio features with our appearance-based representation. As shown in Table 4.5, the fusion result (MA+RNet-(vgg+res)) can further boost the recognition performance (mAP = 87.1) and outperforms previous work. On MED14 dataset, we achieve comparable result (mAP = 36.9) with recent CNN model based methods. Our advantage is that we can simultaneously provide recounting results for event analysis.

4.5. Evaluation results on event recounting

Influence of average pooling. In contrast to the original memory network [34], we add a average pooling layer inside the memory layer, which takes advantage of the spatial organization of the information in the video imprint. Figure 7 demonstrates the influence of adding the average pooling

	Method	mAP	Recounting
CCV	Lai *et al.* [21]	43.6	√
	Jiang *et al.* [19]	59.5	×
	Wu *et al.* [44]	70.6	×
	Nagel *et al.* [25]	71.7	×
	Wu *et al.* [45]	84.9	×
	RNet-(vgg+res)	79.9	√
	MA+RNet-(vgg+res)	**87.1**	√
MED14	IDT [42, 26]	27.6	×
	Gan *et al.* [10]	33.3	√
	Xu *et al.* [47]	36.8	×
	Zha *et al.* [50][3]	**38.7**	×
	RNet-(vgg+res)	36.9	√

Table 4. Comparison with other methods. MA+RNet-(vgg+res) denotes the result fused with audio and motion information using adaptive fusion method [45].

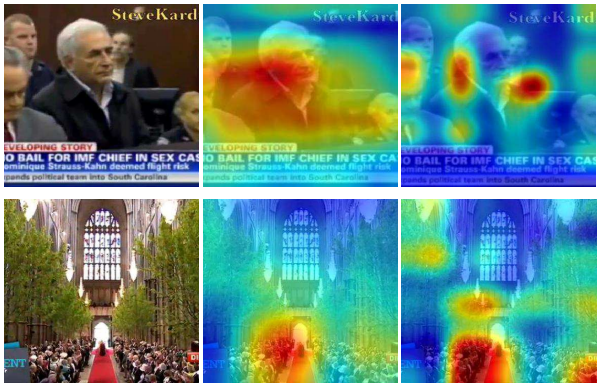

Figure 7. Influence of the average pooling layer in RNet. The middle column shows the recounting map of RNet. The right column shows the recounting map with avg-pooling layer removed.

layer. We can see that the recounting maps are smoother and more reasonable.

Recounting map. Figure 8 illustrates some examples of the recounting results. The heat map is used to visualize the recounting map (the map is rescaled to the same size with the frame). Since no ground truth recounting is available, we can only provide some examples as shown in Figure 8. We can see that our recounting process can not only provide the importance score of each frame, but also indicate the most relevant areas inside each frame. However, due to the coarse resolution of the input feature maps ($\mathbf{S} = 4 \times 4$), the spatial-level recounting results are also very coarse. Nevertheless, the recounting heat map may be treat as a good prior for other post-processing methods, *e.g.*, object segmentation.

[3]Zha *et al.* achieve state of the art result by fusing motion features (IDT) with their CNN based results (mAP = 34.9).

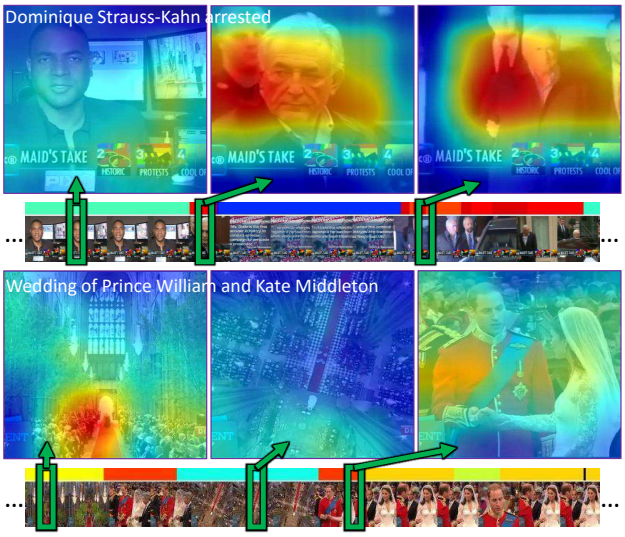

Figure 8. Examples of event recounting results. We use heat map to indicate the recounting map. The key areas related to the event in each frame is painted with red color. The importance score which is computed by the sum of recounting map is shown with color bar (red for important frames) upon the video frame flow.

5. Conclusion and future work

In this paper, we propose a unified framework for complex event retrieval, recognition and recounting. In contrast to previous work, we introduce a feature alignment step to generate the video imprint based on the frame-level features. The feature alignment step can automatically identify and suppress the redundancy across different frames. The experiments show that the video representation generated from the video imprint outperforms previous work both in supervised and unsupervised cases. In addition, with the video imprint, we can further localize the key evidence using the reasoning network. As the followup research, we plan to explore alternative alignment methods which can efficiently handle multiple features and further enhance the video representation. In addition, beyond the content-based video analysis problems, extending the proposed framework to some cross-domain tasks such as video captioning is also a very promising direction.

Acknowledgements

This work was supported by National Key Research and Development Plan 2016YFB1001004, and NSFC Grants 61629301, 61503296, and 61231018. Le Wang was also supported by Fundamental Grant 2015M572563 and the Fundamental Research Funds for the Central Universities Grant XJJ2015066.

References

[1] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *ICCV*, pages 1269–1277, 2015.

[2] M. Baillie and J. M. Jose. Audio-based event detection for sports video. In *International Conference on Image and Video Retrieval*, pages 300–309, 2003.

[3] S. Bhattacharya, M. M. Kalayeh, R. Sukthankar, and M. Shah. Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In *CVPR*, pages 2235–2242, 2014.

[4] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[5] L. Cao, Y. Mu, A. Natsev, S.-F. Chang, G. Hua, and J. R. Smith. Scene aligned pooling for complex video recognition. In *ECCV*, pages 688–701. 2012.

[6] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM TIST*, 2(3):27, 2011.

[7] X. Chang, Y.-L. Yu, Y. Yang, and A. G. Hauptmann. Searching persuasively: Joint event detection and evidence recounting with limited supervision. In *ACM MM*, pages 581–590, 2015.

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

[9] M. Douze, J. Revaud, C. Schmid, and H. Jégou. Stable hyper-pooling and query expansion for event detection. In *ICCV*, pages 1825–1832, 2013.

[10] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, pages 2568–2577, 2015.

[11] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei. You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In *CVPR*, pages 923–932, 2016.

[12] Z. Gao, J. Xue, W. Zhou, S. Pang, and Q. Tian. Democratic diffusion aggregation for image retrieval. *TMM*, 18(8):1661–1674, 2016.

[13] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, pages 392–407, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, June 2016.

[15] H. Jégou and O. Chum. Negative evidences and cooccurences in image retrieval: The benefit of pca and whitening. In *ECCV*, pages 774–787. 2012.

[16] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010.

[17] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah. High-level event recognition in unconstrained videos. *IJMIR*, 2(2):73–101, 2013.

[18] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *arXiv preprint arXiv:1502.07209*, 2015.

[19] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *ICMR*, page 29, 2011.

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

[21] K.-T. Lai, X. Y. Felix, M.-S. Chen, and S.-F. Chang. Video event detection by inferring temporal instance labels. In *CVPR*, pages 2251–2258, 2014.

[22] 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.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[24] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. G. Hauptmann. Complex event detection via multi-source video attributes. In *CVPR*, pages 2627–2633, 2013.

[25] M. Nagel, T. Mensink, C. G. Snoek, et al. Event fisher vectors: Robust encoding visual diversity of visual streams. 2015.

[26] P. Over, J. Fiscus, G. Sanders, D. Joy, M. Michel, G. Awad, A. Smeaton, W. Kraaij, and G. Quénot. TRECVID 2014–an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID*, page 52, 2014.

[27] A. Perina and N. Jojic. Capturing spatial interdependence in image features: the counting grid, an epitomic representation for bags of features. *TPAMI*, 37(12):2374–2387, 2015.

[28] F. Perronnin and D. Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. In *CVPR*, pages 3743–3752, 2015.

[29] J. Revaud, M. Douze, C. Schmid, and H. Jégou. Event retrieval in large video collections with circulant temporal encoding. In *CVPR*, pages 2459–2466, 2013.

[30] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013.

[31] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, pages 806–813, 2014.

[32] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] S. Sukhbaatar, J. Weston, R. Fergus, et al. End-to-end memory networks. In *NIPS*, pages 2440–2448, 2015.

[35] C. Sun and R. Nevatia. ACTIVE: Activity concept transitions in video event classification. In *ICCV*, pages 913–920, 2013.

[36] C. Sun and R. Nevatia. DISCOVER: Discovering important segments for classification of video events and recounting. In *CVPR*, pages 2569–2576, 2014.

[37] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006.

[38] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258, 1997.

[39] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, pages 776–789, 2010.

[40] C.-Y. Tsai, M. L. Alexander, N. Okwara, and J. R. Kender. Highly efficient multimedia event recounting from user semantic preferences. In *ICMR*, page 419, 2014.

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

[42] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[43] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

[44] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue. Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In *ACM MM*, pages 167–176, 2014.

[45] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue. Multi-stream multi-class fusion of deep networks for video classification. In *ACM MM*, pages 791–800, 2016.

[46] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *ACM MM*, pages 461–470, 2015.

[47] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative CNN video representation for event detection. In *CVPR*, pages 1798–1807, 2015.

[48] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation networks for video face recognition. In *CVPR*, 2017.

[49] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, pages 4694–4702, 2015.

[50] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained CNN architectures for unconstrained video classification. In *BMVC*, pages 60.1–60.13, 2015.

[51] Q. Zhang and G. Hua. Multi-view visual recognition of imperfect testing data. In *ACM MM*, pages 561–570, 2015.